

Universidad de Cuenca
Facultad de Ingeniería
Escuela de Informática



“Método para desarrollar sistemas de información de pagos a organizaciones nacionales, permitiendo gestionar facturas electrónicas y enviar información a clientes”

Autores:

Juan Diego Cárdenas Delgado - 0105764062

Edgar Andrés García Clavijo - 0104638507

Director:

Ing. Irene Priscila Cedillo Orellana, PhD. - 0102815842

Trabajo de Titulación Previo a la Obtención del Título de Ingeniero
de Sistemas
Cuenca-Ecuador

2018



Resumen

En las organizaciones las labores de cobranzas juegan un rol sumamente importante para sostener el aspecto financiero de la misma, sin embargo, los mecanismos utilizados para éste propósito se han mantenido en lo tradicional. Por lo cual, las tecnologías de computación en la nube están siendo adoptadas por las organizaciones, motivadas por la necesidad de reducir costos operacionales y proveer servicios escalables y flexibles.

En el área de sistemas de información de pagos y notificaciones existen soluciones de propósitos específicos y limitados, sin embargo no existen soluciones que cubran la gestión de cobranzas mediante notificaciones electrónicas automatizadas sobre información de pagos vencidos a los clientes de las organizaciones. Por este motivo, se ha considerado necesario elaborar un método que permita el desarrollo de sistemas de información que proporcionen agilidad a las organizaciones en la gestión de sus cobranzas.

En este sentido, en el presente trabajo de titulación se propone un método para el desarrollo de sistemas de información de pagos a organizaciones nacionales, que permite gestionar facturas electrónicas y enviar información hacia sus clientes. Este método permite construir una plataforma con tecnologías de procesamiento en la nube, que permita a las organizaciones registrar sus datos y preferencias sobre notificaciones, subir sus facturas e información sobre los pagos pendientes de sus clientes, y realizar la confirmación de dichos pagos. Para la validación del método propuesto se ha realizado la implementación respectiva, dando paso a un caso de estudio que permite analizar la integración y utilización de los servicios provistos por la plataforma desarrollada en una organización nacional.

Palabras Clave: facturación electrónica, computación en la nube, gestión de cobranzas, notificaciones, método, MDA, ASD, seguridad.



Abstract

In organizations, collections work plays an extremely important role in sustaining them financial aspect, however, the mechanisms used for this purpose have remained traditional. Therefore, cloud computing technologies are being adopted by organizations, motivated by the need to reduce operational costs and provide scalable and flexible services.

In the area of information systems of payments and notifications there are solutions for specific and limited purposes, however there are no solutions that cover the collection management through automated electronic notifications about information of due payments to the clients of the organizations. For this reason, it has been considered necessary to develop a method that allows the development of information systems that provide added value to organizations in the management of their collections.

In this sense, the present work proposes a method for the development of systems of information of payments to national organizations, which allows to manage electronic invoices and send information to their customers. This method allows to build a platform with cloud processing technologies, which allows organizations to register their data and preferences about notifications, upload their invoices and information about pending payments of their customers, and make confirmation of those payments. For the validation of the proposed method, the respective implementation has been carried out, giving way to a case study that allows analyzing the integration and use of the services provided by the platform developed in a national organization.

Key Words: e-invoicing, cloud computing, collection management, notifications, method, MDA, ASD, security.



Contenido

Resumen	2
Abstract	3
Contenido	4
Índice de figuras	7
Índice de tablas	9
Cláusula para publicación en repositorio	10
Clausula propiedad intelectual	11
Cláusula para publicación en repositorio	12
Clausula propiedad intelectual	13
Agradecimientos	14
Dedicatoria	15
Dedicatoria	16
Capítulo 1	17
Introducción	17
1.1 Motivación y contexto	17
1.2 Planteamiento del problema	21
1.3 Solución propuesta	22
1.4 Hipótesis y objetivos	23
1.4.1 Hipótesis	23
1.4.2 Objetivos	23
1.4.3 Objetivo general	23
1.4.4 Objetivos específicos	23
1.5 Metodología de la investigación	24
1.6 Estructura del trabajo	25
1.7 Aporte científico	27
Capítulo 2	28
Marco teórico / tecnológico	28
2.1 Revisiones sistemáticas	28
2.2 Facturación electrónica (e-invoicing)	29
2.3 Desarrollo Adaptable de Software (ASD)	29
2.4 Tecnologías de ingeniería basadas en modelos	30
2.4.1 Desarrollo de software dirigido por modelos	30
2.4.2 Ingeniería basada por modelos	30
2.4.3 Arquitectura dirigida por modelos	30
2.5 Análisis de recursos utilizados en la elaboración del método	31
2.5.1 Metodologías de desarrollo con MDA	31
2.5.2 Comparativa de metodologías MDA	37
2.5.3 Metodologías de desarrollo ágil	38
2.5.4 Comparativa de metodologías ágiles	43



2.6	I-Estrella (i*)	45
2.7	Sistemas de información web	45
2.8	Meta modelo de ingeniería de procesos de software	45
2.9	Modelo y notación de procesos de negocio	46
2.10	Certificados X.509	46
2.11	Autoridad de certificación	46
2.12	Firma electrónica	47
2.13	Tecnologías involucradas	47
1.	Marco de desarrollo de modelado Eclipse	47
2.	Servicios web	47
3.	Computación en la nube	47
4.	Glassfish	48
5.	Java	48
6.	MySQL	48
7.	Protocolo de acceso simple a objetos	48
8.	Protocolo de transferencia de hipertexto	48
9.	POST	48
10.	Protocolo simple de transferencia de correo	49
11.	Servicio de mensajes cortos	49
12.	Correo electrónico	49
13.	Amazon Web Services	49
Capítulo 3		50
Estado del arte		50
3.1	Método de estudio	50
3.1.1	Etapas de planificación	50
3.1.1.1	Establecer las preguntas de investigación	50
3.1.1.2	Fuente de datos y estrategia de búsqueda	51
3.1.1.3	Cadena de búsqueda	51
3.1.1.4	Periodo de búsqueda	52
3.1.1.5	Criterios de extracción	52
3.1.2	Etapas de ejecución	53
3.1.2.1	Selección de estudios primarios	53
3.1.2.2	Evaluación de la calidad	54
3.1.2.3	Método de análisis y síntesis	54
3.1.3	Etapas de reporte de la revisión	57
Capítulo 4		64
Método para desarrollo de sistemas de información de pago		64
4.1	Contexto	64
4.2	Metodologías seleccionadas	64
4.3	Estructura del método	64
4.3.1	Vista general	65
4.4	Descripción del método	65
4.4.1	Fase de inicio del proyecto	65
4.4.1.1	Análisis temprano de requerimientos	67
4.4.1.2	Identificación de riesgos	68
4.4.2	Fase de análisis y requerimientos	68
4.4.2.1	Elicitación de requerimientos	68
4.4.2.2	Análisis y documentación de requerimientos	69
4.4.3	Fase de soporte MDA	72
4.4.3.1	Identificación y especificación del entorno de desarrollo y lenguaje de modelado	72
4.4.3.2	Identificación y especificación de modelos necesarios	73



4.4.3.3	Identificación y especificación de transformaciones y herramientas	73
4.4.3.4	Modelado	74
4.4.4	Fase de desarrollo adaptable	74
4.4.4.1	Especulación	76
4.4.4.2	Colaboración	76
4.4.4.3	Aprendizaje	77
Capítulo 5		79
Implementación		79
5.1	Fase Inicial	79
5.1.1	Requerimientos tempranos	79
5.1.2	Propósito y elicitación	79
5.1.3	Creación del modelo	79
5.1.3.1	Contexto del sistema	79
5.1.3.2	Diagramas de modelado I-Estrella	82
5.2	Fase de análisis y requerimientos	87
5.2.1	Descripción de modelo SR, I-Estrella	87
5.2.2	Escenarios	92
5.2.3	Casos de uso	95
5.2.4	Diagramas de secuencia	97
5.2.5	Diagramas de estado	101
5.2.6	Arquitectura	103
5.3	Fase de soporte MDA	107
5.3.1	Especificar entorno de desarrollo, lenguaje de modelado	107
5.3.2	Modelos necesarios	107
5.3.3	Identificaciones y selección de transformaciones y herramientas	108
5.3.4	Modelado	108
5.4	Fase de desarrollo adaptable	110
5.4.1	Especulación	110
5.4.2	Colaboración	111
5.4.3	Aprendizaje	121
Capítulo 6		124
Validación mediante caso de uso		124
6.1	Diseño y planeación de caso de estudio	124
6.1.1	Definir el caso de estudio	124
6.2	Reelección de datos	125
6.3	Análisis de datos recolectados	127
6.4	Discusión y conclusiones del caso de estudio	131
Capítulo 7		132
Conclusiones y trabajos futuros		132
Referencias		134
Anexos		144
1.	Anexo A: Símbolo, Términos y Descripción de SPEM 2.0	144
2.	Anexo B: Símbolo, Tipo y Descripción de BPMN	145
3.	Anexo C: Símbolo, Tipo y Descripción de i*	146
4.	Anexo D: Oficio para caso de estudio en GSoft	148
5.	Anexo E: Entrevistas del caso de estudio	149
6.	Anexo F: Artículos de investigación 1	155
7.	Anexo G: Artículos de investigación 2	156



Índice de figuras

Figura 1-1 - Interacción en sistemas de información de gestión empresarial (Delgado & Marín, 2000) ...	17
Figura 1-2: Esquema de la solución propuesta.	23
Figura 1-3: Metodología de la investigación.	25
Figura 1-4: Estructura del trabajo de titulación.	26
Figura 2-1: Marco metodológico MIDAS (Cáceres & Marcos, 2002).	34
Figura 2-2: Metodología ASD (James A. Highsmith & Orr, 2001).	38
Figura 2-3: Metodología DSDM (Voigt, 2004).	40
Figura 2-4: Metodología FDD (Pekka Abrahamsson et al., 2002).	41
Figura 2-5: Iteraciones FDD (Pekka Abrahamsson et al., 2002).	42
Figura 3-1: Porcentaje de estudios correspondiente a EC3: Tipos de instituciones.	55
Figura 3-2: Porcentaje de estudios correspondientes a EC4: Partes que incluyen.	55
Figura 3-3: Porcentaje de estudios correspondientes a EC11: Atributos de seguridad.	56
Figura 3-4: Porcentaje de estudios correspondientes a EC12: Como se abordan los estudios.	56
Figura 3-5: Comparación entre EC3: Tipo de instituciones y EC4: Partes que incluye.	58
Figura 3-6: Comparación entre EC11: Atributos de seguridad, EC3: Tipos de institución y EC4: Partes que incluyen.	58
Figura 3-7: Distribución demográfica por países.	63
Figura 4-1: Vista general de método.	66
Figura 4-2: Fase de inicio de proyecto.	67
Figura 4-3: Fase de análisis y requerimientos.	69
Figura 4-4: Fase de soporte MDA.	72
Figura 4-5: Fase de desarrollo adaptable.	74
Figura 4-6: Proceso ASD (James A. Highsmith & Orr, 2001).	76
Figura 5-1: Dependencia Plataforma – Organización.	84
Figura 5-2: Dependencia Plataforma - Servidor de Correo.	84
Figura 5-3: Dependencia Plataforma - Pasarela de SMS.	85
Figura 5-4: Dependencia Plataforma - Autoridad de certificación.	85
Figura 5-5: Dependencia Pasarela de SMS - Cliente.	86
Figura 5-6: Dependencia Servidor de Correo - Cliente.	86
Figura 5-7: Diagrama general SD.	87
Figura 5-8: Diagrama SR Plataforma – Organización.	88
Figura 5-9: Diagrama SR Plataforma - Servidor de Correo.	89
Figura 5-10: Diagrama SR Plataforma - Pasarela de SMS.	90
Figura 5-11: Diagrama SR Plataforma – Autoridad de certificación.	90
Figura 5-12: Diagrama SR Cliente – Pasarela de SMS.	91
Figura 5-13: Diagrama SR Cliente - Servidor de Correo.	92
Figura 5-14: Diagrama BPMN del registro y actualización de datos de la organización.	93
Figura 5-15: Diagrama BPMN de la recepción de comprobantes para notificación.	94
Figura 5-16: Diagrama BPMN de la confirmación de comprobantes pagados.	94
Figura 5-17: Diagrama BPMN del envío de notificaciones de correo y SMS.	95
Figura 5-18: Registro y actualización de datos.	98
Figura 5-19: Recepción de comprobantes.	99
Figura 5-20: Confirmación de comprobantes pagados.	100
Figura 5-21: Envío de notificaciones.	101
Figura 5-22: Registro o actualización de datos de la organización.	102
Figura 5-23: Envío de notificaciones.	102
Figura 5-24: Confirmación de comprobantes pagados.	102
Figura 5-25: Arquitectura de la plataforma.	106
Figura 5-26: Modelo independiente de plataforma.	109
Figura 5-27: Modelo específico de plataforma.	110
Figura 5-28: Transformación hacia capa de lógica de negocio.	112
Figura 5-29: Transformación hacia implementación DAO, capa de persistencia.	112



<i>Figura 5-30: Transformación hacia hilos notificadores.</i>	<i>113</i>
<i>Figura 5-31: Capa lógica de negocio (clases del modelo).....</i>	<i>113</i>
<i>Figura 5-32: Ejecución de la instancia.</i>	<i>121</i>
<i>Figura 5-33: Capa de persistencia (Clases).</i>	<i>122</i>
<i>Figura 6-1: Abstracción de datos (Runeson & Höst, 2009).</i>	<i>127</i>
<i>Figura 6-2: Tendencia de días transcurridos en pagos.</i>	<i>129</i>
<i>Figura 6-3: Suspensión de servicios, antes de la plataforma.....</i>	<i>130</i>
<i>Figura 6-4: Suspensión de servicio después de la plataforma.</i>	<i>130</i>



Índice de tablas

<i>Tabla 1-A: (SRI, 2017)</i>	20
<i>Tabla 2-A: Comparativa metodologías MDA (Chitforoush et al., 2007).</i>	37
<i>Tabla 2-B: Comparativa metodologías ágiles (Pekka Abrahamsson et al., 2002).</i>	45
<i>Tabla 3-A: Fuentes de búsqueda manual.</i>	51
<i>Tabla 3-B: Cadena de búsqueda.</i>	52
<i>Tabla 3-C: Criterios de extracción.</i>	53
<i>Tabla 3-D: Número de artículos incluidos.</i>	54
<i>Tabla 3-E: Evaluación de calidad.</i>	54
<i>Tabla 3-F: Estudios por criterios de extracción.</i>	62
<i>Tabla 5-A: Actores del sistema.</i>	83
<i>Tabla 5-B: Caso de uso para enviar datos y plantillas de empresa.</i>	96
<i>Tabla 5-C: Caso de uso para enviar notificaciones.</i>	96
<i>Tabla 5-D: Caso de uso para la confirmación de Pago(s).</i>	97
<i>Tabla 5-E: Iteración 1.</i>	111
<i>Tabla 5-F: Iteración 2.</i>	111
<i>Tabla 5-G: Iteración 3.</i>	111
<i>Tabla 5-H: Iteración 4.</i>	111
<i>Tabla 6-A: Síntesis de resultados (Entrevistas).</i>	128
<i>Tabla 6-B: Información de pagos antes de la plataforma.</i>	129
<i>Tabla 6-C: Información de pagos después de la plataforma.</i>	129
<i>Tabla 6-D: Tiempos estimados para notificación.</i>	131



Cláusula para publicación en repositorio



Universidad de Cuenca

Cláusula de licencia y autorización para publicación en el Repositorio Institucional

Juan Diego Cárdenas Delgado en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación *"Método para desarrollar sistemas de información de pagos a organizaciones nacionales, permitiendo gestionar facturas electrónicas y enviar información a clientes"*, de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, abril de 2018

Juan Diego Cárdenas Delgado

C.I: 0105764062



Clausula propiedad intelectual



Universidad de Cuenca
Clausula de propiedad intelectual

Juan Diego Cárdenas Delgado, autor de la tesis *"Método para desarrollar sistemas de información de pagos a organizaciones nacionales, permitiendo gestionar facturas electrónicas y enviar información a clientes"*, certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, abril de 2018

Juan Diego Cárdenas Delgado

C.I: 0105764062



Cláusula para publicación en repositorio



Universidad de Cuenca

Cláusula de licencia y autorización para publicación en el Repositorio Institucional

Yo, Edgar Andrés García Clavijo en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación *"Método para desarrollar sistemas de información de pagos a organizaciones nacionales, permitiendo gestionar facturas electrónicas y enviar información a clientes"*, de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, Abril de 2018

Edgar Andrés García Clavijo

C.I: 0104638507



Clausula propiedad intelectual



Universidad de Cuenca
Clausula de propiedad intelectual

Yo, *Edgar Andrés García Clavijo*, autor de la tesis "Método para desarrollar sistemas de información de pagos a organizaciones nacionales, permitiendo gestionar facturas electrónicas y enviar información a clientes", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, Abril de 2018

Edgar Andrés García Clavijo

C.I: 0104638507



Agradecimientos

El presente trabajo de titulación representa el resultado de una etapa de trabajo de quienes han conformado el equipo de investigación y han colaborado para la consecución de los objetivos planteados. En este sentido, queremos agradecer a la Ing. Priscila Cedillo Orellana PhD. por su apoyo y dedicación al brindarnos directrices y conocimientos para el desarrollo de este trabajo.

De igual manera, agradecemos a los docentes de la universidad que nos han formado académicamente en las distinguidas aulas de la Universidad de Cuenca, compartiéndonos sus conocimientos que han resultado de gran utilidad en este trabajo.

También queremos agradecer a la Ing. Alexandra Bermeo, por su colaboración en la realización de los artículos científicos que han surgido con este trabajo. Finalmente, expresamos nuestro agradecimiento a quienes conforman GSoft por brindarnos información y permitirnos realizar la aplicación del caso de estudio.

Juan Diego, Andrés



Dedicatoria

Este trabajo de titulación dedico a quienes forman parte de mi vida y me han apoyado incondicionalmente, a esas personas por las cuales es posible esto y han estado de una u otra manera en el cumplimiento de esta meta.

Quienes me han formado y visto crecer, a mis padres, gracias a su amor, enseñanzas y valores me han hecho ser el gran ser humano que soy; a mis hermanos, que han estado siempre conmigo brindándome consejos siguiendo su gran ejemplo.

Finalmente y no menos importante a dos personas quienes son una parte esencial de mi vida y han estado a mi lado cada día brindándome su amor y comprensión; gracias a mi esposa y compañera de vida que me ha sabido apoyar con sus palabras de aliento en el cumplimiento de este objetivo, así mismo a mi hija quien es la motivación para seguir adelante, por quien lucho cada día para así lograr ser un referente positivo para ella.

Juan Diego Cárdenas Delgado



Dedicatoria

Quiero dedicar este trabajo de titulación a aquellas personas que me han dado la fuerza, el apoyo y me han mostrado el camino para hacer posibles mis sueños. A mi abuelito Ricardo, quien siempre estuvo pendiente por mis estudios. A mis padres Edgar y Miriam, que han sido mi apoyo incondicional y me han acompañado en todo momento, brindándome sus enseñanzas, su cariño y su ejemplo para perseverar cada día y conseguir mis objetivos. A mi hermano Juda, quien siempre ha estado conmigo alegrándome la vida, por quien persevero en la búsqueda de ser mejor cada día para apoyarlo en su camino. A Diana, Rómulo, Isaac, Elías, Froilán y Valentina, que han estado siempre pendientes de mí y dispuestos a brindar su apoyo y cariño. A mis amigos y todas las personas que supieron apoyarme y de una u otra forma ayudaron a la culminación de este trabajo de titulación.

Edgar Andrés García Clavijo

Capítulo 1

Introducción

En este capítulo se presenta la motivación, contexto, problemática actual y la solución propuesta en el presente trabajo de titulación, incluyendo la hipótesis, objetivos generales y específicos. Adicionalmente, se muestran los aportes científicos generados con el desarrollo de este trabajo.

1.1 Motivación y contexto

En la actualidad existen sistemas de información de gestión empresarial que se aplican a la gestión de procesos de distinta índole dentro de las organizaciones, siendo elementos fundamentales en la gestión de cadena de suministro; es decir, el intercambio de información y material por los agentes involucrados en un canal logístico, desde la materia prima hasta que un producto es terminado. La evolución tecnológica en dichos sistemas ha permitido conectar los sistemas de gestión de cada empresa, en lo que se conoce como soluciones negocio a negocio (*B2B - Bussiness to Bussiness*) y negocio a consumidor, (*B2C - Bussiness to Consumer*), por sus siglas en inglés, respectivamente. En este sentido, los sistemas *B2B* tienen como misión mejorar la relación entre las empresas, utilizando sistemas de información compartidos. Los sistemas *B2C* están enfocados a la interacción con clientes finales utilizando tecnologías de comunicación modernas (Delgado & Marín, 2000). La interacción entre dichas soluciones está representada en la Figura 1-1.

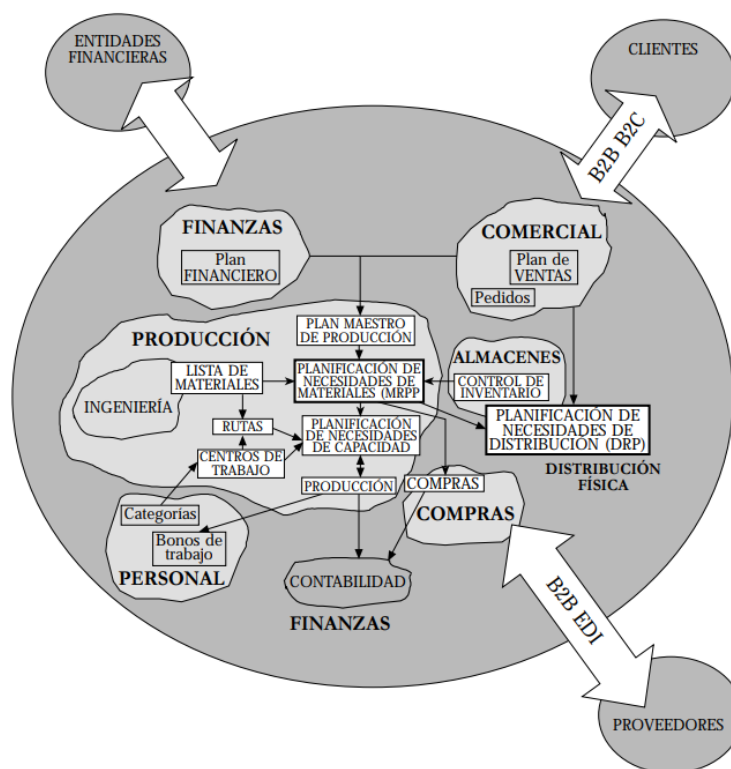


Figura 1-1 - Interacción en sistemas de información de gestión empresarial (Delgado & Marín, 2000)

Dentro de la gestión comercial de las organizaciones se encuentran los sistemas de información de pago. Por lo general, estos sistemas son utilizados de forma exclusiva por cada una de las organizaciones, manteniendo completa independencia en el envío de información de pagos hacia sus clientes, en una relación *B2C*.

Por otro lado, los avances de las Tecnologías de la Información y Comunicaciones (TIC) han permitido el desarrollo de sistemas de información que incluyen más funcionalidades, dando paso a una paulatina migración de los medios de entrega de información a los clientes, desde medios convencionales (p. ej., mensajería tradicional, fax) hacia los medios digitales (p. ej., mensajes de texto, correo electrónico). Los beneficios percibidos al utilizar medios electrónicos favorecen a la disminución de los tiempos de entrega, ahorros en costos relacionados a la impresión de documentos y entrega a domicilio, facilidad de manejo de la logística de entrega y seguimiento, mayor seguridad en el resguardo de la información y cuidado del medio ambiente (Rivera & Pereira, 2011).

Una de las principales preocupaciones de las organizaciones es la recuperación de cartera, para favorecer a la recuperación de su liquidez (Romero, Corrigan, & Mccauley, 2018); de ahí, ha emergido la necesidad de medios de cobro electrónicos e información de cuentas vencidas a los clientes de las organizaciones. En el Ecuador, el Servicio de Rentas Internas (SRI) es la institución estatal encargada de gestionar las políticas tributarias, siguiendo el marco de los principios constitucionales, asegurando la suficiencia recaudatoria cuyo destino es fomentar la cohesión social (SRI, 2018b). Desde el año 2015, el SRI ha iniciado la inclusión del esquema de comprobantes electrónicos (facturas, retenciones, notas de débito, notas de crédito) de propósito mercantil, dando paso a una migración paulatina de los distintos tipos de contribuyentes, pasando de comprobantes físicos a comprobantes electrónicos (SRI, 2013). De la misma manera, una nueva resolución emitida por el SRI, incluye a un nuevo grupo de contribuyentes para la migración hacia comprobantes electrónicos (SRI, 2017). Este aspecto se establece en los artículos expuestos en dicha resolución, como se detalla a continuación en la Tabla 1-A:

Artículo 1. “Están obligados a emitir facturas, comprobantes de retención, guías de remisión, notas de crédito y notas de débito, a través de mensajes de datos y firmados electrónicamente, los siguientes sujetos pasivos”, (SRI, 2017):

Tipo de contribuyente	Condición	Cumplimiento
Proveedores del estado	Facturación en el ejercicio fiscal anterior (2017), sea ésta igual o superior al valor de ingresos establecidos para estar obligados a llevar contabilidad	Ingresos anuales del ejercicio fiscal inmediato anterior, superior a 15 Fracción Básica Excedente (FBE)
Personas naturales o sociedades proveedores de exportadores	Facturación en el ejercicio fiscal anterior igual o superior al valor de ingresos establecido para estar obligados a llevar contabilidad, y que al menos el 50% de la misma corresponda a transacciones efectuadas directamente con exportadores	Ingresos anuales del ejercicio fiscal inmediato anterior, superior a 15 FBE
Personas naturales o sociedades	Ingresos anuales del ejercicio fiscal anterior sean iguales o superiores al monto contemplado para pequeñas y medianas empresas	Desde \$100.001,00 hasta \$5'000.000,00



Personas naturales o sociedades	Sujetos pasivos del Impuesto a los Consumos Especiales (ICE)	Art. 80 LORTI
Personas naturales o sociedades	Sujetos pasivos del Impuesto Redimible a las Botellas Plásticas no Retornables (TBPNR)	Artículos agregados a continuación del artículo 89 LORTI
Personas naturales o sociedades	Quienes soliciten al Servicio de Rentas Internas devolución del impuesto Redimible a las Botellas Plásticas no Retornables IBPNR	Art. 2 No. NAC-DGERCGC16-00000470
Personas naturales o sociedades	Productos y comercializadores de alcohol	
Personas naturales o sociedades	Titulares de concesiones mineras, contratos de explotación minera, licencias y permisos, así como de las autorizaciones para instalar y operar plantas de beneficio, fundición y refinación, y de licencias de comercialización; en general, todos los titulares de derechos mineros, excluidos los que se encuentran bajo el régimen de minería artesanal	
Personas naturales o sociedades	Contribuyentes que desarrollen la actividad económica de elaboración y refinado de azúcar de caña	
Personas naturales o sociedades	Importadores habituales de bienes	Cumplir con las dos condiciones: -Importaciones netas iguales o mayores al 25% de compras netas anuales del ejercicio anterior (2017) y -Ingresos brutos del ejercicio anterior iguales o superiores a 15 FBE. Vigente desde Julio/18
Personas naturales o sociedades	Agentes Afianzados de Aduana	Vigente desde Julio/18
Personas naturales o sociedades	Presten Servicio de transporte de valores y especies monetarias	Servicio prestado a través de vehículos blindados. Vigente desde Julio/18
Personas naturales o sociedades	Venta, Cesión o Transferencia de cartera a cualquier título	Vigente desde Julio/18
Sociedades	Consortios, alianzas estratégicas (cuyos miembros sean facturadores electrónicos)	
Personas naturales o sociedades	Cuyos ingresos anuales del ejercicio fiscal anterior sean iguales o superiores al monto contemplado para pequeñas empresas, señaladas en el literal c) del artículo 1 de esta Resolución, obligación de emisión	Desde \$100.001,00 hasta \$1.000.000,00



	electrónica de facturas, comprobantes de retención, guías de remisión, notas de crédito y notas de débito, será exigible a partir del 01 de enero de 2019.	
--	--	--

Tabla 1-A: (SRI, 2017)

Este aspecto ha abierto la posibilidad de disponer de toda la información registrada de dichos comprobantes en un ordenador. Adicionalmente, se han definido estándares para estos comprobantes de manera que puedan ser procesados por cualquier sistema de información conforme las necesidades de cada organización.

En el presente trabajo de titulación se propone un método para el desarrollo de sistemas de información de pagos, que permita procesar las facturas electrónicas emitidas por las organizaciones y enviar notificaciones con información de las deudas que mantienen los clientes con las compañías, utilizando medios digitales. Este método puede ser utilizado por organizaciones públicas o privadas, incluso a nivel de gobierno, para la creación de soluciones que promuevan la utilización de medios digitales para notificaciones hacia los clientes de las compañías.

Por otro lado, la computación en la nube se define como un modelo que permite el acceso de red ubicuo, oportuno y bajo demanda, hacia un conjunto de recursos computacionales compartidos que son provistos y liberados rápidamente con mínimo esfuerzo administrativo o intervención del proveedor de servicios (Mell & Grance, 2011). La utilización de una infraestructura orientada a servicios tiene algunas ventajas, como interoperabilidad en la independencia de tecnologías de lenguaje de programación, integración sencilla de sistemas, reúso de código fuente asociado a un único desarrollo para un gran número de usuarios en la web (Cavanaugh, 2006). En este sentido, en el presente trabajo de titulación se ha planteado una solución a ser desplegada en ambientes que hacen uso de las tecnologías de computación en la nube, teniendo en cuenta todas las características específicas de este tipo de plataformas y siguiendo un modelo orientado a servicios, por las ventajas que esto conlleva (alta disponibilidad, reducción de costos, modos de facturación, entre otros). De esta manera, se presenta una solución eficiente y efectiva, que tenga como objetivo el facilitar las operaciones de cobranzas generando eficiencia para las organizaciones y creando cercanía con sus clientes gracias a las notificaciones que les ayudan a mantenerse pendientes sobre sus pagos.

Para la consecución de los objetivos de este trabajo de titulación, han intervenido metodologías de desarrollo ágiles y el enfoque de arquitecturas dirigidas por modelos (*MDA - Model Driven Architecture*); todo esto se ha planificado por los beneficios que se obtienen en el proceso de desarrollo de software. *MDA* permite a los desarrolladores enfocarse en el diseño de alto nivel, sin considerar restricciones tecnológicas o implementaciones específicas. Esto conlleva ventajas en cuanto a mantenibilidad frente a cambios tecnológicos, portabilidad entre plataformas y facilidad de documentación (Martínez, Lamoth, Moreno, & Jacho, 2015).

La validación del método propuesto en este trabajo de titulación ha sido llevada a cabo utilizando la metodología propuesta por Runeson y Host (2009) para la ejecución y documentación de casos de estudio de ingeniería de software, la cual presenta prácticas recomendadas y listas de verificación descubiertas y evaluadas empíricamente basándose en la experiencia de investigadores.



Para ello se ha realizado un caso de estudio en la organización local “GSoft”, el cual ha permitido analizar de manera general los aspectos positivos y negativos encontrados en las labores de cobranza de la organización, tras la implementación de la plataforma en la nube y su integración en la organización.

1.2 Planteamiento del problema

Las organizaciones tienen una gran tarea y carga administrativa cuando se trata de gestionar sus cobranzas, debido a que en su mayoría, el cobro de las cuentas vencidas dificulta una operación ágil de las organizaciones, ya que para realizar esta gestión existe una gran demanda de recursos utilizados, tales como: tiempo, recurso humano, monetario, etc. Por este mismo hecho, las organizaciones en la actualidad buscan alinearse con las nuevas tecnologías para optimizar sus recursos y procesos de manera eficiente. La finalidad de incluir la tecnología en este accionar empresarial, se enfoca en muchas ocasiones en la entrega de información sobre la situación crediticia del cliente a través de una notificación electrónica.

Hoy en día, existen varias soluciones que envían notificaciones por medio del correo electrónico y la mensajería de texto, como es el caso de Riofrio Ramires (2011), el cual propone un módulo de notificaciones para el aviso de vencimiento de pagos de compañías de seguro en el entorno ecuatoriano; sin embargo, su solución está orientada únicamente a avisos de pago de pólizas de seguro, sin considerar la integración de otros tipos de organizaciones, ni la utilización de comprobantes electrónicos. Por otro lado, Peña Aguirre (2016) muestra la integración de un módulo a una plataforma ERP de una compañía de Bogotá, Colombia para el envío de correos electrónicos masivos. Sin embargo, esta solución no contempla la normativa ecuatoriana y carece de la gestión y utilización de comprobantes electrónicos.

Consecuentemente, se ha determinado como un problema, el que en Ecuador no existe una plataforma la cual envíe notificaciones de pago de las organizaciones hacia sus clientes, haciendo uso de comprobantes sujetos a las normativas impuestas por el SRI (lo que facilitaría la extracción de información y evitaría procesos de migración complejos); de ahí, se propone el planteamiento de un método y su posterior instanciación a fin de determinar su factibilidad, utilizando los beneficios de *MDA* para definir modelos independientes de plataforma (*PIM*) y proponer reglas de mapeo a modelos específicos de plataforma (*PSM*); ya que de lo que se conoce, ninguna plataforma en la actualidad informa a los clientes sobre sus facturas a pagar a través de los diferentes medios digitales y de uso cotidiano como lo son los mensajes de texto, e-mails, entre otros.

Con respecto a las plataformas que utilizan *MDA* se tiene a Muller, Studer, Fondement, y Bezivin (2005), quienes tratan sobre un modelado y desarrollo de aplicaciones web independientes de la plataforma, introduciendo un modelo específico para el modelado de aspectos dinámicos específicos de la web. La aplicación web puede ajustarse a distintos patrones de diseño, entre los cuales, se tendrá en cuenta el modelo vista controlador (MVC). Dadas las restricciones que la especificación de un modelo conceptual presenta, se utilizan lenguajes de restricción de objetos (*OCL* – Object Constraint Language), los cuales agregan mayor semántica y restricciones al modelo en cuestión, lo que permite que se ajuste de mejor manera a las necesidades del negocio. Por otro lado, Raghupathi y Umar (2008)



conceptualizan y desarrollan una arquitectura orientada a modelos para sistemas de salud clínica con el fin de realizar seguimiento de información de los pacientes. Se propone un modelo generalizado al cual se aplican reglas de transformación *PIM* a *PSM* para generar prototipos de la aplicación. Finalmente, en el trabajo de Cáceres, Marcos, Vela, y Rey (2008), se propone una metodología orientada a modelos basados en *MDA* para el desarrollo de sistemas de información Web (*WIS*). En donde se aplica un metamodelo *MDA* a la plataforma web utilizando *XML* y tecnologías objeto relacionales. Por otro lado, también se proponen modelos independientes de la plataforma (*PIM*) y modelos específicos de plataforma (*PSM*), definiendo algunas reglas de transformación entre modelos. Al final se presenta un caso de estudio que ilustra la propuesta. Cabe destacar que si bien estas soluciones no abarcan el dominio que se estará abordando en este trabajo, éstas representan soluciones guía que son de gran utilidad durante la consecución de la solución aquí propuesta.

En conclusión, el presente trabajo de titulación está centrado específicamente en la realización de un método para el desarrollo de plataformas de pagos para los clientes de diferentes organizaciones haciendo uso de las facturas electrónicas que se vienen manejado en Ecuador desde hace varios años, para que puedan recibir alertas de las mismas. Finalmente, se puede acotar que el (*OMG* - Object Management Group) ha reconocido que el enfoque de modelado es una manera muy robusta para la especificación de sistemas, y así lo demuestra con *MDA*, la misma que busca brindar al diseñador la posibilidad de expresar la estructura, comportamiento y funcionalidad del sistema independientemente de los aspectos tecnológicos, logrando también una alta interoperabilidad.

1.3 Solución propuesta

Dada la necesidad de un método que aporte en el desarrollo de sistemas de información de notificaciones para facilitar las labores de cobranza en las organizaciones, que permitan enviar mensajes a sus clientes a través de medios digitales y que se ajusten a la normativa ecuatoriana en la gestión de comprobantes electrónicos, en este trabajo de titulación se plantea la creación de un método que permita desarrollar una plataforma utilizando tecnologías de computación en la nube, la cual gestione facturas electrónicas de las organizaciones nacionales y envíe información de pago a sus clientes por medio de notificaciones electrónicas, generando eficiencia en las tareas de cobranza para la recuperación de cartera. En la Figura 1-2 se muestra un esquema sencillo sobre la solución propuesta.



Figura 1-2: Esquema de la solución propuesta.

1.4 Hipótesis y objetivos

Se definen a continuación las hipótesis y los objetivos.

1.4.1 Hipótesis

H₀: El método propuesto, no presenta utilidad para el desarrollo de sistemas de información de pagos a organizaciones nacionales, que gestionen facturas electrónicas y envíen información de pago a sus clientes a través de medios digitales.

H_a: El método propuesto, resulta de utilidad en el desarrollo de sistemas de información de pagos a organizaciones nacionales, que gestionen facturas electrónicas y envíen información de pago a sus clientes a través de medios digitales; además proporciona agilidad a las organizaciones en su gestión de cobranzas para recuperación de cartera.

1.4.2 Objetivos

El desarrollo de este trabajo de titulación se ha basado en los siguientes objetivos generales y específicos:

1.4.3 Objetivo general

Proponer un método, para el desarrollo de sistemas de información de pagos a organizaciones ajustado a la realidad y normativa ecuatoriana, que haga uso de arquitecturas dirigidas por modelos.

1.4.4 Objetivos específicos

- Definir de manera clara los requerimientos de las organizaciones y delimitar el alcance.
- Recabar y analizar información sobre plataformas desarrolladas mediante un enfoque *MDA*.
- Establecer los aspectos tecnológicos a considerar, incluyendo arquitecturas, estándares, lenguajes de programación y herramientas.



- Definir el método y establecer cada una de las actividades involucradas en la misma.
- Definir la lógica de negocio basado en un modelo independiente de computación junto al análisis de requerimientos y proponer un modelo independiente de la plataforma.
- Transformar el modelo independiente a un modelo específico de plataforma y realizar la integración.
- Realizar una implementación de una plataforma conforme el método planteado y realizar las validaciones necesarias a través de un caso de estudio.

1.5 Metodología de la investigación

Para la elaboración de la propuesta, se ha seguido una metodología estructurada conforme al modelo de transferencia tecnológica de Gorschek, Garre, Larsson, y Wohlin (2006), el cual contempla ocho actividades que pretenden encontrar una solución realista mediante un proceso iterativo de validación empírica de soluciones candidatas, como se muestra en la Figura 1-3 y se detallan a continuación:

1. Análisis del problema: consiste en comprender el problema que da origen y propósito a la investigación, para lo cual se observa el dominio y se identifican las necesidades de la industria o las empresas.
2. Formulación del problema: luego de analizar el problema, se lo formula de forma clara y precisa, incluyendo factores de contexto, objetivos que la investigación persigue, planteamiento de preguntas de investigación y justificación del estudio.
3. Revisión del estado del arte: por medio de una revisión sistemática de literatura se esclarece el estado del arte, revisando soluciones existentes e identificando brechas que la investigación desea abordar.
4. Solución candidata: se propone una solución al problema establecido, por medio de un método definido.
5. Entrenamiento: consiste en una actividad de tipo incremental, donde se busca proporcionar el conocimiento necesario a los profesionales del área para generar una vista general de la aplicación de la solución propuesta.
6. Validación inicial: la validación de la solución propuesta se la realiza en un entorno de laboratorio, para lo cual se utiliza un caso de estudio.
7. Validación realista: se aplica un entorno real industrial para llevar a cabo casos de estudio o experimentos controlados.
8. Liberación de la solución: Se hace una valoración de los resultados obtenidos y se preparan las herramientas y material requerido para su uso.

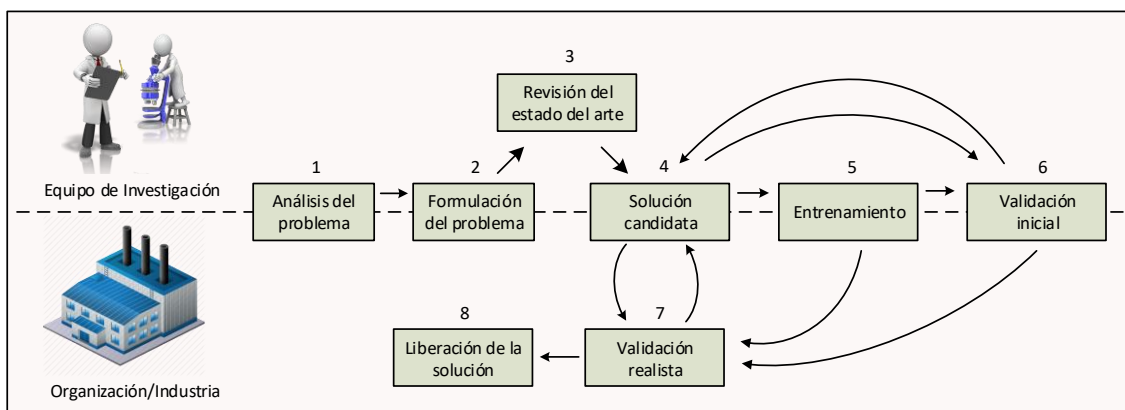


Figura 1-3: Metodología de la investigación.

En este trabajo se han cubierto las primeras seis actividades de este modelo, proponiendo como trabajos futuros las actividades de validación realista y liberación de la solución.

1.6 Estructura del trabajo

A continuación, se muestra la estructura de este trabajo de titulación con una breve descripción de cada capítulo. La Figura 1-4 muestra un esquema de dicha estructura, relacionando las actividades de la metodología de Gorschek et al. (2006) con cada capítulo.

- Capítulo 2: Marco teórico

Se presenta un compendio de descripciones y nociones clave sobre tecnologías, protocolos, herramientas y conceptos teóricos utilizados para elaborar este trabajo, como facturación electrónica, revisión sistemática, desarrollo de software adaptable (ASD – Adaptive Software Development), MDA, servicios web, protocolos web, certificados, entre otros.

- Capítulo 3: Estado del arte

Se muestra el proceso y los resultados de la aplicación de la metodología de Kitchenham y Charters (2007), para esclarecer el estado del arte, por medio de una revisión sistemática de literatura, centrada en “facturación electrónica, plataformas y notificaciones”.

- Capítulo 4: Método para desarrollo de sistemas de información de pago

Se describe a detalle el método propuesto, mostrando sus fases, tareas, roles involucrados en cada una de éstas, entradas y salidas de cada proceso.

- Capítulo 5: Implementación

Se presenta la implementación de una plataforma, haciendo uso del método propuesto en el presente trabajo.

- Capítulo 6: Validación mediante caso de uso

Se realiza la validación de la implementación del método propuesto en el capítulo 4, para lo cual se utiliza la metodología de Runeson y Höst (2009), aplicada a la implementación realizada en el capítulo 5.

- Capítulo 7: Conclusiones y trabajos futuros

Se presenta la información más relevante obtenida con el desarrollo de este trabajo. Adicionalmente se proponen trabajos futuros relacionados al presente.

- Anexos

Contiene toda la documentación adjunta y generada en el transcurso del presente trabajo de titulación, como son: artículos de investigación aceptados en diferentes conferencias, nomenclaturas y otros documentos generados durante la implementación del método en el caso de estudio y la validación del mismo.

Es importante recalcar que las actividades de validación realista y liberación de la solución de la metodología seguida para el desarrollo de esta propuesta, se vislumbran como trabajos futuros cuando se tenga acceso a validaciones y experimentación empírica en diferentes empresas de Ecuador. En este sentido, se ha considerado conveniente empezar con una validación local (Cuenca), con lo cual se esclarezca el contexto de investigación para una posterior extensión a otras ciudades.

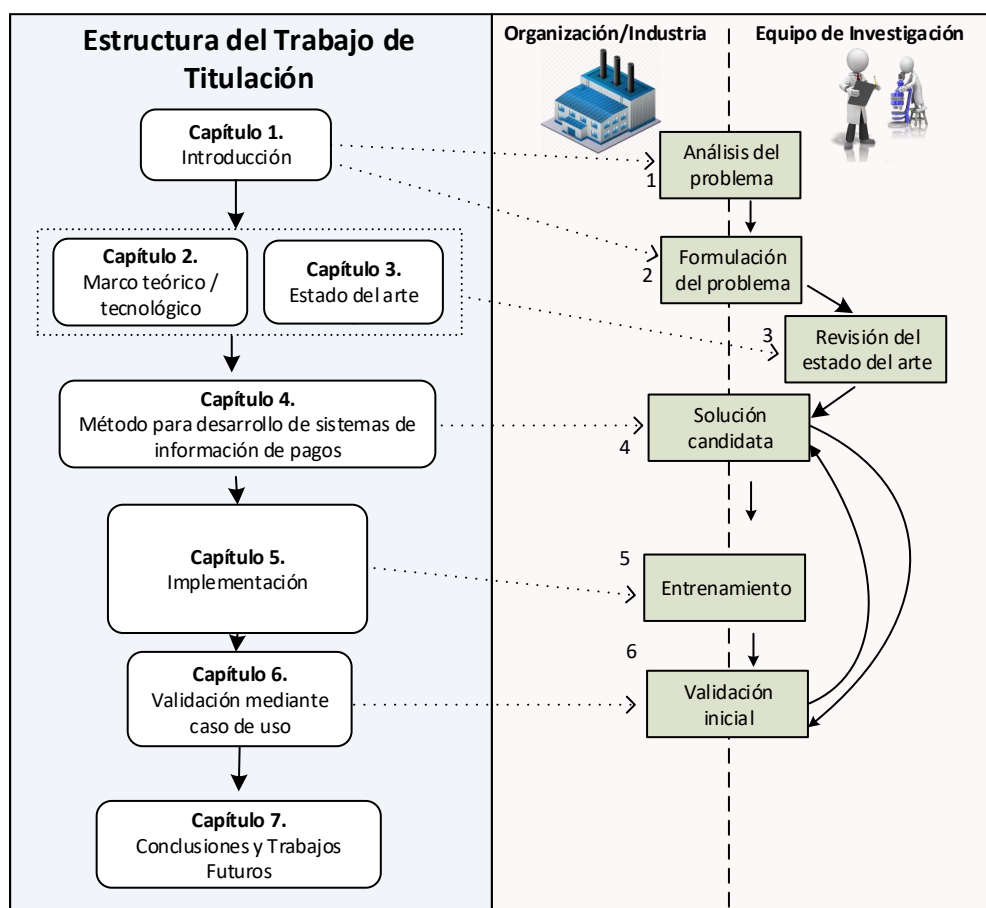


Figura 1-4: Estructura del trabajo de titulación.



1.7 Aporte científico

El presente trabajo de titulación muestra su aporte científico mediante dos artículos de difusión científica, el uno presentado en una conferencia internacional y el otro ha sido elaborado y enviado, en espera de respuesta por parte de pares académicos de la conferencia:

- “A systematic literature review of electronic invoicing, platforms and notification systems” aprobado para presentación en la 5th International Conference on eDemocracy & eGovernment - ICEDEG 2018 con sede en Ambato, Ecuador a llevarse a cabo los días 4 al 6 de abril de 2018 (García & Cárdenas, 2018) .
- “Architecture for Electronic Notifications Services on the Cloud” enviado y en espera de aceptación (Cárdenas & García, 2018).



Capítulo 2

Marco teórico / tecnológico

En el presente capítulo se plasma el marco teórico o tecnológico, en donde se cita los diferentes conceptos que son abordados a lo largo de dicho trabajo de titulación. Con el fin de fundamentar las bases para un correcto entendimiento del mismo. A continuación, se presenta un compendio de conceptos, tecnologías y elementos que son utilizados o están relacionados a este trabajo de titulación.

2.1 Revisiones sistemáticas

Una revisión sistemática de literatura es un método que permite evaluar e interpretar todas las investigaciones relevantes a una interrogante de investigación específica, área temática o fenómeno de interés. Estas revisiones presentan una evaluación de un tópico de investigación utilizando una metodología rigurosa, auditable y confiable.

Kitchenham y Charters (2007) propone lineamientos derivados de tres directrices existentes utilizados por investigadores médicos, dos libros publicados por investigadores con intereses en ciencias sociales, y debates con investigadores de otras disciplinas relacionadas a las prácticas basadas en evidencia. Estos lineamientos se han adaptado para reflejar los problemas o necesidades específicas en la investigación de ingeniería de software, proponiendo tres fases comunes en revisiones sistemáticas: planeación de la revisión, ejecución de la revisión y documentación de la revisión.

Los objetivos asociados a planeación de la revisión son:

- Identificar la necesidad de una revisión
- Identificar claramente el propósito de la revisión.
- Especificar interrogantes o preguntas de investigación
- Desarrollar un protocolo de revisión
- Evaluar el protocolo de revisión

Los objetivos asociados con la ejecución de la revisión son:

- Identificar la investigación
- Seleccionar estudios primarios
- Evaluar la calidad de los estudios primarios
- Extraer los datos y monitorearlos
- Sintetizar los datos extraídos

Los objetivos asociados con la documentación de la revisión son:

- Especificar mecanismos de disseminación
- Conformar el reporte principal



- Evaluar el reporte

2.2 Facturación electrónica (e-invoicing)

El concepto de facturación electrónica se originó en Escandinavia. Las facturas en un formato estándar son emitidas, transmitidas y recibidas de forma electrónica a través del Internet. Éstas son transmitidas entre propietarios de negocios, permitiendo un ahorro de costos y mejorando la eficiencia operacional. Los sistemas de facturación electrónica permiten a las organizaciones evitar el uso de papel, protegiendo al medio ambiente e incrementando la eficiencia. Los países como Estados Unidos, Finlandia, Dinamarca, Suecia y Bélgica, han acordado promover su utilización a nivel de gobierno. Taiwán es uno de los países pioneros en la región Asia-Pacífico. La facturación electrónica tiene beneficios incluyendo aspectos de seguridad anti fraude (Lian, 2015).

En Ecuador, el SRI es la institución pública encargada de las gestiones de política tributaria, basándose en la constitución y sus principios, para conseguir la suficiencia recaudatoria y fomentar la cohesión social (SRI, 2018a). En este contexto, la facturación electrónica es una forma alternativa de emitir comprobantes de venta, cumpliendo con los requerimientos reglamentarios y legales que se exigen para que el SRI pueda autorizarlos, garantizando autenticidad de origen y contenido íntegro mediante el mecanismo de firma electrónica (SRI, 2018a, 2018b, 2018c).

2.3 Desarrollo Adaptable de Software (ASD)

Es una metodología ágil para la implementación y desarrollo de software. De la misma manera que otras metodologías ágiles, tiene un funcionamiento cíclico, y en cada iteración se realizan cambios y adaptaciones. Por ello se dice que en lugar de luchar se adapta a los cambios, implementando patrones ágiles para el desarrollo. En cuanto a su ciclo de vida ASD propone tres fases esenciales, especular-colaborar-aprender, mientras que lo común en las metodologías es la planificación-diseño-construcción (J.A Highsmith, 2000). ASD realiza estimaciones de tiempo, con la premisa de que se pueden extender o desviar. Sin embargo, se vuelve algo necesario, ya que de esta manera se logra priorizar las diferentes tareas para tratar de centrarse en el tiempo (James A. Highsmith & Orr, 2001).

James A. Highsmith y Orr (2001) proponen el siguiente ciclo de vida:

- Especular
Es una fase de iniciación del proyecto, en donde se lleva a cabo una planificación tentativa del proyecto. De esta manera se establecen los objetivos principales, metas del proyecto, alcance y limitaciones del mismo. De la misma manera se establece el número de iteraciones para realizar el proyecto.
- Colaborar
Esta fase está centrada en un desarrollo por ciclos. En la cual el equipo colabora intensamente para realizar las planificaciones anteriormente definidas y se construyen las diferentes funcionalidades como un componente o grupo de entregables a ser desarrollados en un ciclo iterativo.
- Aprender
Es esta fase se realiza un control de calidad, el cual permite un aprendizaje individual, brindando al equipo una mayor posibilidad de éxito. De esta manera,



mejora el entendimiento sobre las tecnologías, los procesos utilizados y el proyecto. Se analizan cuatro categorías: (I) *Revisión de calidad desde la perspectiva del cliente*, (II) *Revisión de calidad desde una perspectiva técnica*, (III) *Rendimiento del equipo de desarrollo y de las prácticas utilizadas* y (IV) *Estado del proyecto*.

2.4 Tecnologías de ingeniería basadas en modelos

2.4.1 Desarrollo de software dirigido por modelos

El desarrollo de software dirigido por modelos (DSDM), (*MDD* - Model Driven Development) por sus siglas en inglés, es una propuesta para el desarrollo de software en la que se les atribuye un papel principal de todo el proceso a los modelos. DSDM busca aumentar en nivel de abstracción en el desarrollo de software, de tal forma convierte a los modelos y a las transformaciones entre ellos en los principales artefactos de todas las fases de proceso de desarrollo de software: captura y gestión de los requisitos, diseño, análisis, implementación, despliegue, configuración, mantenimiento, evolución, entre otros (Pons, Giandini, & Pérez, 2010). La transformación entre modelos constituye el motor de DSDM. En cuanto a sus puntos clave se tiene:

- La abstracción a mayor nivel en la especificación tanto del problema a resolver como de la solución planteada mediante los modelos y sus relaciones.
- La automatización genera mayor confianza, ya que es asistida por las diferentes herramientas de modelado para soportar el análisis, diseño y la ejecución.
- El uso de estándares industriales como medio para facilitar las comunicaciones, la interacción entre diferentes aplicaciones y productos, y la especialización tecnológica.

2.4.2 Ingeniería basada por modelos

La ingeniería dirigida por modelos (*MDE* – Model Driven Engineering), es un paradigma de ingeniería de software, la cual se basa en el modelado como una de sus principales actividades para el ciclo de vida completo de un proyecto. Plantea la idea central de que los artefactos fundamentales son los modelos para el desarrollo de software, más no los programas (Cuadrado, Cánovas Izquierdo, & Molina, 2014). De esta manera tiene como objetivo aumentar la productividad maximizando la compatibilidad entre sistemas, mediante la reutilización de los modelos estandarizados. Así mismo simplifica el proceso de diseño y promueve la comunicación entre individuos y equipos que trabajan en el sistema (Rodrigues Da Silva, 2015). Una de las iniciativas de *MDE* más conocidas que establece el consorcio (*OMG* - Object Management Group), es la Arquitectura Dirigida por Modelos (*MDA* – Model Driven Architecture) (Kent, 2002), la cual ha sido concebida para dar soporte a *MDE*.

2.4.3 Arquitectura dirigida por modelos

Es una arquitectura propuesta por (*MOF* - Meta Object Facility), la cual utiliza meta modelos especificados en (*UML* – Unified Modeling Language) y proporciona la base fundamental de guías para estructurar especificaciones expresadas como modelos. Unifica todos los pasos del desarrollo y la integración desde un modelo de negocio, modelo

arquitectónico y de aplicación, hasta el desarrollo, despliegue, mantenimiento y evolución. En base a los meta modelos que proporciona se puede describir los lenguajes de modelado como objeto interrelacionados. De esta manera los estándares relacionados especifican el intercambio de modelos usando una estructura *XML* específica, llamada *XMI*. Una de las claves de *MDA* es la capacidad de transformación de los modelos Modelo a Modelo (*M2M* – Model To Model) y Modelo a Texto (*M2T* – Model To Text), utilizando (*QVT* - Query View Transform) que proporciona los mecanismos para dichas transformaciones (Chitforoush, Yazdandoost, & Ramsin, 2007). De esta manera *MDA* ofrece varios beneficios como:

- **Productividad**
Es posible gracias a los diferentes modelos que *MDA* ofrece, como lo es, el modelo independiente de cómputo (*CIM*), modelo independiente de plataforma (*PIM*) y modelo específico de plataforma (*PSM*). De esta manera dichos modelos son capaces de realizar las transformaciones automáticamente, al menos en gran parte, al igual que la generación de código. Así, el trabajo lo realiza la herramienta y no el desarrollador.
- **Portabilidad**
Gracias a que dispone de un modelo el cual es independiente de la plataforma, todo lo definido en este modelo es portable hacia cualquier plataforma.
- **Interoperabilidad**
Esta propiedad lo logra gracias a los modelos *PSM*, los cuales no podrán comunicarse directamente entre ellos, ya que pueden pertenecer a distintas tecnologías. Este problema lo resuelve generando no solo los modelos *PSM*, sino los puentes entre ellos.
- **Mantenimiento y documentación**
Básicamente el modelo *PIM* es el que desempeña el papel de la documentación, ya que se encuentra definido a un alto nivel de abstracción, el cual se necesita para cualquier sistema de software.

En *MDA*, se manejan los conceptos de *CIM*, *PSM* y *PIM*. El modelo de computación independiente (*CIM*) muestra un modelo de dominio comercial; mientras que el modelo independiente de plataforma (*PIM*) muestra un modelo de solución de software que no contiene detalles de la plataforma. De manera similar, el modelo específico de plataforma (*PSM*) muestra el modelo detallado de la plataforma que se implementará (OMG, 2018).

2.5 Análisis de recursos utilizados en la elaboración del método

En esta sección se muestra los diferentes recursos, los cuales fueron profundizados para obtener los más adecuados para el método. De esta forma, se seleccionaron los más apropiados para la plantación del método propuesto.

2.5.1 Metodologías de desarrollo con *MDA*

MODA TEL

MODA TEL consiste en un proceso de desarrollo de software basado en los principios y conceptos de *MDA*. Está especializado para aplicaciones distribuidas, sin embargo puede ser aplicable a otros dominios y situaciones. *MODA TEL* está definido de acuerdo a (SPEM -

Software Process Engineering Meta-model). En esta metodología se da una separación de actividades de preparación y ejecución: (Chitforoush et al., 2007)

- Administración del proyecto

Durante esta fase el proceso de desarrollo de software es descrito en términos de sus actividades, luego estas actividades son relacionadas con roles y procedimientos para aseguramiento de calidad.

- Preparación preliminar

El objetivo de esta fase es identificar necesidades de modelo y transformación. Se expresa la ejecución final del sistema como una plataforma abstracta, definiendo el lenguaje de modelado apropiado y otras necesidades específicas.

- Preparación detallada

Los modelos y sus transformaciones son especificados en esta fase, basándose en los lenguajes de modelado y necesidades definidas en la fase anterior.

- Configuración de la infraestructura

En esta fase se seleccionan las herramientas que den soporte a las actividades de desarrollo con *MDA*, como herramientas de generación de código automático o semiautomático.

- Ejecución del proyecto

Comprende la ejecución de las actividades de desarrollo, que dependen del proceso de desarrollo seleccionado en la fase de administración de proyectos. Se consideran siete actividades de esta fase: análisis de requerimientos, modelado, verificación y validación, transformación, codificación y pruebas, integración y despliegue, y operación y mantenimiento.

Si bien esta metodología tiene soporte para *MDA*, y cubre un proceso de ciclo de vida genérico, no se considera pues está más orientada a sistemas distribuidos cuyo proceso de desarrollo se aborda con un ciclo de vida secuencial y concentrada la mayoría de actividades únicamente en una fase de toda la metodología.

MASTER

Es un proyecto europeo de instrumentación (IST), en donde se ha propuesto una metodología basada en *MDA*. Este proceso consiste en ocho fases principales las cuales consisten en varias actividades (Chitforoush et al., 2007).

- Captura de los requisitos de usuario

En esta fase el objetivo es obtener y documentar los requisitos del cliente, obteniendo como resultado un modelo de aplicación en donde se formalizan los requisitos de cliente, un *PIM* inicial y especificación de requisitos funcionales iniciales.

- Definición del contexto *PIM*



Aquí se definen los objetivos y alcance del sistema. Así como se identifican los actores externos, especifica los principales servicios ofrecidos por el sistema.

- Especificación de requisitos *PIM*

El objetivo principal es refinar el contexto *PIM*, así como especificar casos de uso e identificar requisitos no funcionales.

- Análisis *PIM*

Aquí las funcionalidades del sistema y aspectos de calidad se describen con vistas al interior del sistema y se realiza una trazabilidad a los requisitos *PIM*.

- Diseño

En esta fase primero se realiza un modelo independiente de la plataforma para todos los requisitos. Luego se refina para indicar la solución específica de la plataforma.

- Codificación e integración

Basados en el enfoque ideal de *MDA*, el código debe producirse automáticamente desde el *PSM* a través de las transformaciones.

- Pruebas

Los casos de prueba deben generarse automáticamente a partir del modelo de prueba (que es un refinamiento del *PIM*) a través de los motores de transformación.

- Despliegue

En esta fase se entrega al cliente el sistema desarrollado.

MIDAS

Es un marco metodológico, el cual está orientado a modelos para la elaboración y desarrollo de sistemas de información web (*WIS*) con tecnología *XML*. Para los modelos propuestos se agrupan en una dimensión estructural y otra de comportamiento, mientras que el límite entre los modelos independiente de la plataforma y específicos se fija en el diseño lógico. *MIDAS* en lugar de introducir un proceso específico, se enfoca en tres dimensiones para el modelado de *WIS*: 1) niveles, que consulte el contenido de hipertexto y los niveles de presentación, 2) fases, que se refieren a las fases del ciclo de vida del software, y 3) aspectos, que se refieren a la estructura y puntos de vista de modelado conductual (Chitforoush et al., 2007)(Cáceres & Marcos, 2002). De esta manera se muestra en marco en la Figura 2-1.

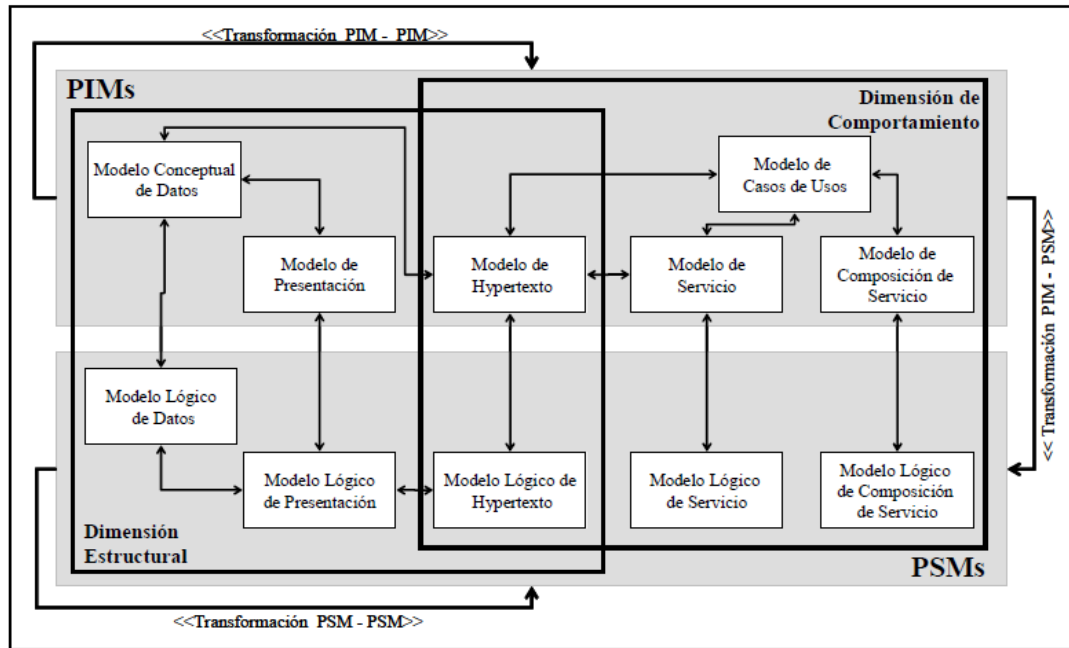


Figura 2-1: Marco metodológico MIDAS (Cáceres & Marcos, 2002).

C3

Es un proceso de desarrollo de software que está integrado en una metodología concurrente, colaborativa y basada en componentes enriquecida con técnicas *MDD*. En cuanto a las características principales de C3 se tiene a su repositorio de dominio que contiene metadatos y componentes específicos del dominio. Su proceso consiste en dos fases principales (Chitforoush et al., 2007):

- Fase de estandarización: Esta fase consulta el repositorio de metadatos de dominio, meta modelos y ontologías. Esto provee una interfaz para buscar los elementos de software del dominio, de forma que se puedan descargar en el repositorio local del proyecto. De igual manera, los nuevos modelos de componentes o de dominio específico pueden ser subidos y categorizados para utilización en futuros proyectos.
- Fase de desarrollo de software: Aquí se realizan tres pasos principales y comunes en la mayoría de metodologías: Diseño de modelos, Generación de código, Despliegue de aplicaciones.

- Diseño de modelos

Los desarrolladores de componentes y equipos de desarrollo trabajan de forma colaborativa en la arquitectura de la aplicación de negocio, representada con el estándar *XMI*. Se utiliza un repositorio de manera que el equipo de administración y supervisión del proyecto pueda proporcionar retroalimentación.

- Generación de código

Las facetas de la aplicación de negocio pueden ser modeladas con *UML* o *XMI*, en un nivel independiente de la plataforma. Se utilizan herramientas de generación de código para transformar los modelos en componentes de software desplegable, específico de la plataforma.



- Despliegue de aplicación

Se completa el código faltante de forma manual, los componentes de software deben estar listos para el despliegue en un servidor de aplicaciones, conforme el modelo arquitectónico.

ODAC

Esta metodología está basada en el modelo de referencia de procesamiento distribuido abierto RM-ODP propuesto en ISO/IEC (1998), con el potencial de ser una metodología orientada a *MDA* (Chitforoush et al., 2007). Consiste en tres fases principales:

- Análisis
Se da la especificación de comportamiento, utilizando un modelo independiente de la plataforma (*PIM*).
- Diseño
Se da la especificación de ingeniería con un modelo dependiente de la plataforma que describe el entorno de ejecución.
- Implementación
Se realiza la especificación operacional, un modelo específico de plataforma que es el resultado de la transformación del *PIM* al *PDM*.

Metodología MDA de Chitforoush

Dadas las características de orientación a objetos y desarrollo basado en componentes de *MDA*, se puede complementarlo con procesos de desarrollo de software existentes. Las generalidades de flexibilidad y capacidades de abstracción de *MDA* facilitan la integración con las metodologías existentes. La metodología propuesta por Chitforoush et al. (2007) puede ser utilizada para aumentar los procesos de desarrollo existentes, incluyendo conceptos *MDA*, principios y tecnologías. Se proponen cuatro fases junto con actividades umbrellita que cubren todo el proyecto para complementar dichas fases:

- Fase de iniciación del proyecto
Los objetivos, tamaño y alcance del proyecto son identificados. Se da una exploración e identificación de los riesgos y restricciones, personas, organizaciones y sistemas externos. Se asignan miembros del equipo de trabajo incluyendo expertos del dominio y expertos *MDA*.
- Fase de análisis y selección del proceso de desarrollo de software
Los requerimientos especificados son usados como base para seleccionar un proceso de desarrollo de software (*SDP*), que debe ser adaptado para cubrir los requerimientos, documentando una descripción de las modificaciones necesarias para el proceso en términos de las actividades que deben ser realizadas.
- Fase de soporte *MDA*
Esta fase puede ser ejecutada de forma paralela con la fase previa. Se toman decisiones acerca del desarrollo orientado a modelos del sistema y los aspectos arquitecturales de alto nivel. Para esto se contemplan las siguientes actividades:
 - Identificación y especificación de plataforma

Los arquitectos del sistema identifican la plataforma final hacia la cual el sistema será implementado y desplegado, esto incluye aspectos tecnológicos de hardware y software. Para esto se utiliza un modelo de plataforma (*PM*), que será utilizado posteriormente en la transformación *PIM* a *PSM*.

- Identificación y especificación de lenguaje de modelado

Se identifican diferentes perspectivas del sistema a ser modelado. Es decir, con la presencia de los expertos de dominio y expertos *MDA* se decide si modelar la funcionalidad, comportamiento o estructura en modelos de caso de uso, diagramas de estado y diagramas de clase, respectivamente. Dependiendo de estos modelos se selecciona un lenguaje apropiado que permita expresar el dominio y la situación del proyecto.

- Identificación y especificación de transformaciones

En esta actividad se identifican transformaciones necesarias entre modelos. El principal enfoque se da en transformaciones de *PIMs* a *PSMs*, pero de ser necesario también se consideran transformaciones entre *PIMs* o *PSMs*. Una vez decidido esto, se debe especificar en detalle las reglas de transformación y anotaciones. Se pueden utilizar transformaciones estándares ya conocidas, como *UML-to-J2EE*.

- Selección de herramientas

Para seleccionar las herramientas *MDA* se debe tener en cuenta que numerosas actividades serán realizadas por dichas herramientas: transformación de modelos, generación de código, aplicación de reglas y anotaciones.

- Fase SDP de ejecución

Es la fase principal del proyecto donde se desarrollan los productos finales del proyecto. Las actividades específicas de esta fase dependen de la metodología seleccionada en la segunda fase. Posterior a al análisis de Chitforoush et al. (2007) se ha observado que las metodologías ágiles *Crystal Clear*, *FDD* y *ASD* pueden ser integradas en este entorno de trabajo sin mayor modificación de sus procesos. Por ejemplo al considerar *ASD* las modificaciones que se realizan se aplican al ciclo de desarrollo iterativo, haciendo que las etapas del proceso *ASD* sean compatibles con el enfoque *MDA*. El proceso resultante contemplaría las siguientes actividades en la etapa de desarrollo de componentes: modelado, verificación, transformación de modelos, codificación y pruebas, integración y despliegue.

- Actividades umbrella

Estas actividades corresponden a la monitorización y administración aplicadas en paralelo a todas las fases mencionadas. Esto incluye dirección de proyectos, control de calidad, gestión de riesgos y entrenamiento. Es importante considerar la administración de reuso, pues el desarrollo orientado a modelos tiene gran énfasis en el reuso de modelos, lenguajes de modelado y especificaciones de transformación.

2.5.2 Comparativa de metodologías MDA

El análisis basado en criterios propuesto en Chitforoush et al. (2007) sobre las metodologías revisadas, se muestra en la Tabla 2-A. A continuación se describen los criterios analizados:

- **Soporte MDA:** La capacidad de la metodología para soportar conceptos y prácticas MDA. Todas las metodologías analizadas se basan en MDA, sin embargo utilizan conceptos, principios y técnicas diferentes. Por ejemplo, C3 utiliza estándares MDA como UML y XMI, junto con transformaciones dentro del proceso, pero no aborda tipos de modelo como PIM, PSM o PM.
- **Inclusión de proceso:** Establece si la metodología incorpora o no un proceso específico y concreto; existen varias metodologías que afirman basarse en MDA, pero no prescriben un proceso concreto, alegando que MDA satisface sus necesidades de proceso.
- **Cobertura del ciclo de vida genérico:** Hace referencia a la medida en que una metodología cubre las fases y actividades del ciclo de vida genérico de desarrollo de software.
- **Precisión del proceso:** Si el proceso se encuentra detallado y suficientemente preciso para ser seguido de forma eficiente y efectiva por los desarrolladores. Por ejemplo, MIDAS y MASTER no propone actividades exactas en sus procesos.

Alcance de la aplicación: Es el alcance de los proyectos y dominios a los que se aplica la metodología, algunos metodologías propuestas no son lo suficientemente generales para que se puedan aplicar a todo proyecto. A continuación se muestra una comparativa de las metodologías MDA en la Tabla 2-A.

Metodología	Soporte MDA	Inclusión de procesos	Cobertura del ciclo de vida genérico	Precisión de procesos	Alcance de la aplicación
MODA-TEL	Completo	Si	Si	Alto	Si
MASTER	Completo	Si	Si	Bajo	Si
MIDAS	Completo	Parcial	No	Muy bajo	No (Sistemas de información web)
C3	Parcial (Transformaciones de modelo, estándar MDA)	Parcial	Parcial	Bajo	No (Software de aplicación comercial y activos de software de dominio específico)
ODAC	Completo	Si	Parcial	Bajo	Si
Metodología propuesta por Chitforoush	Completo	Si	Si	Alto	Si

Tabla 2-A: Comparativa metodologías MDA (Chitforoush et al., 2007).

2.5.3 Metodologías de desarrollo ágil

SCRUM

Es una metodología desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle, en donde define un marco para la gestión de proyectos, la misma se centra en actividades de tipo gerenciales y no especifica prácticas de ingeniería. Así mismo motiva a la generación de equipos auto dirigidos cooperativos, aplicando revisiones frecuentes a manera de control. Es por ello que Scrum funciona de manera correcta cuando los requisitos están perfectamente definidos y en donde la variabilidad de los mismos es mínima. De esta manera Scrum es ideal para proyecto en donde los requerimientos son inestables, ya que fomenta el surgimiento de los mismo (Ordoñez, Suarez, & Villavicencio, 2016) (Ramírez & Flórez, 2014).

En cuanto a las características se puede englobar, por un lado el desarrollo de software mediante iteraciones, denominadas sprints, y por otro lado las reuniones que se tienen a lo largo del proyecto (Letelier, Canós, Sánchez, & Penadés, 2003). Además de ciertos elementos como los equipos auto dirigidos, una colaboración cercana por parte del cliente, modelos incremental en cuanto al desarrollo con entregas funcionales y sobre todo la parte de motivación y responsabilidades por parte de los equipos por la autogestión, auto-organización y compromiso (Ramírez & Flórez, 2014).

Desarrollo adaptable de software

El desarrollo de software adaptable (ASD - Adaptive Software Development) fue propuesto por Jim Highsmith y Sam Bayer, es una metodología que en lugar de luchar se adapta a los cambios implementando patrones ágiles para el desarrollo. En cuanto a los ciclos de vida a diferencia de otras metodologías que tienen planificaciones-diseño-construcción, en ASD se proponen tres fases esenciales, dichas fases se denomina especular-colaborar-aprender (J.A Highsmith, 2000).

En cuanto a la fase de especular, es en donde se da inicio al proyecto y además se realiza la planificación en cuanto a las características del software; en la segunda fase se da como tal el desarrollo de las características y para finalizar en la tercera fase se realiza una revisión en cuanto a la calidad, cabe recalcar que es un ciclo iterativo en donde se revisan los componentes, y con esto se corrigen los errores cometidos para volver a iniciar el ciclo de desarrollo (Letelier et al., 2003). En la Figura 2-2 se puede observar cómo se procede en el ciclo de vida de la metodología.

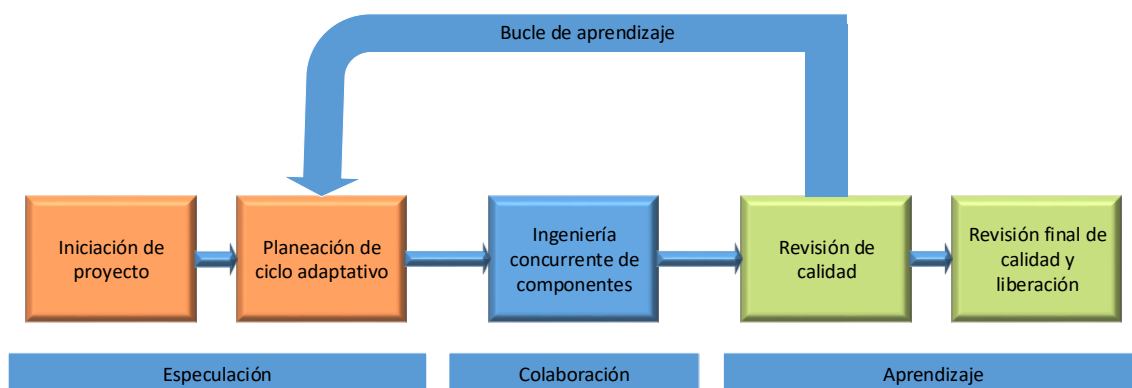


Figura 2-2: Metodología ASD (James A. Highsmith & Orr, 2001).



En cuanto a las ventajas se puede mencionar y destacar a la tercera fase, en donde se realiza la revisión de componentes, la misma que sirve para aprender de los errores. Así mismo dichos cambios generan información útil para el mejoramiento del comportamiento del software. Por otro lado apunta hacia el desarrollo rápido de aplicaciones (*RAD* - Rapid Application Development), el cual enfatiza la velocidad de producto para obtener un producto de alta calidad (J.A Highsmith, 2000).

Crystal family

La familia de metodologías *Crystal* consiste en múltiples metodologías y principios que pueden ser ajustados a proyectos de distinta índole. Dichos métodos consisten en colores transparente, amarillo, naranja y rojo, basándose en que el color más oscuro corresponde a la metodología más pesada. De esta manera, dependiendo del tamaño y criticidad del proyecto se debe escoger una metodología del color apropiado (P Abrahamsson, Warsta, Siponen, & Ronkainen, 2003).

Por otro lado, existen reglas y características que son comunes en la familia *Crystal*. Los proyectos siempre utilizan ciclos de desarrollo incremental con una duración entre uno y tres meses, en los cuales se intensifica la comunicación y cooperación de la gente. Los métodos *Crystal* utilizan como insumos los resultados de (*XP* - *Extreme Programming*), como casos de uso, borradores de pantallas y bosquejos de diseño. También están abiertos a la utilización en conjunto con otras prácticas de desarrollo, herramientas o insumos, tales como *XP* o *SCRUM* (Stojanovic, Dahanayake, & Sol, 2003).

La metodología más liviana es *Crystal Clear*, que propone una forma altamente optimizada para usarse con equipos pequeños de desarrollo, priorizando el aseguramiento de una entrega satisfactoria, eficiencia en el desarrollo y entorno de las asambleas de trabajo. Puede ser descrita de forma breve de la siguiente manera: “El diseñador principal con dos hasta siete desarrolladores, en un lugar amplio o cuartos adyacentes, utilizando plasmadores de información como pizarras y rotafolios, con acceso sencillo a expertos del dominio o usuarios, manteniendo alejadas las distracciones, entregando código ejecutándose, probado y usable a los usuarios, cada mes o dos, reflejando y ajustando sus asambleas de trabajo de forma periódica (Clear & Cockburn, 2014).

Método de desarrollo de sistemas dinámicos

El método de desarrollo de sistemas dinámicos (*DSDM* – Dynamic System Development Method) es un marco de trabajo para el desarrollo de software de forma ágil. Como se detalla en Voigt (2004), éste método consta de fases y subfases claramente definidas, roles y principios que permiten a los equipos de desarrolladores trabajar eficientemente. Estas fases se muestran a continuación en la Figura 2-3:

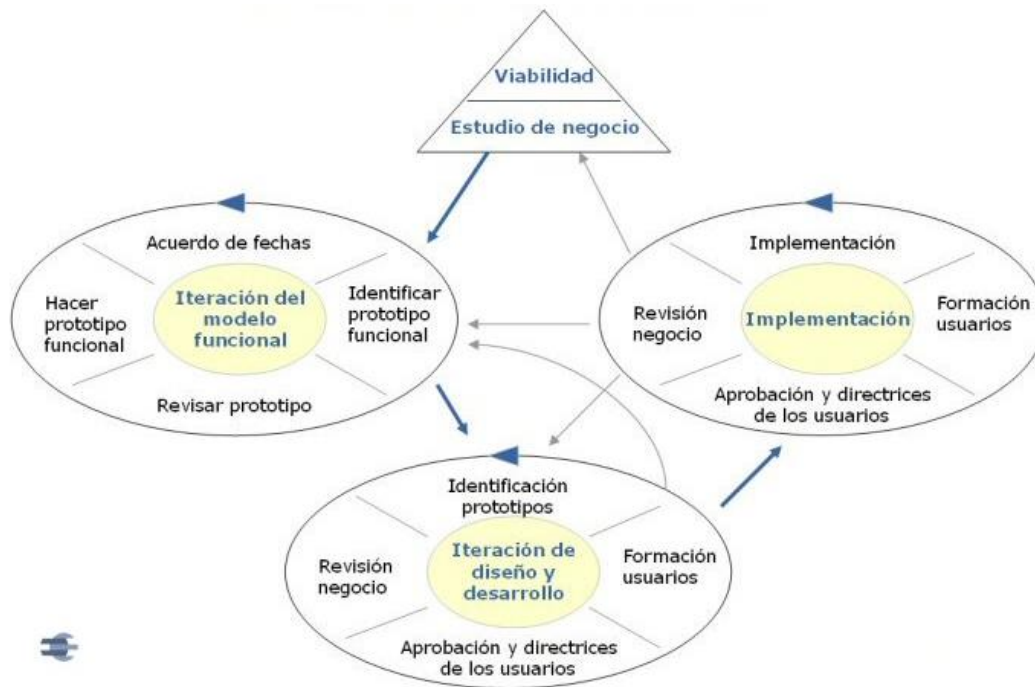


Figura 2-3: Metodología DSDM (Voigt, 2004).

- Pre-proyecto: Esta fase incluye sugerencias del proyecto y selección de proyectos candidatos. Aquí se determina si un proyecto debería llevarse a cabo.
- Estudio de factibilidad: Se realiza la definición del problema a abordar, evaluaciones de costos probables y factibilidad técnica.
- Estudio de negocio: Provee la base para el trabajo subsecuente. Es tan corto como sea posible, con duración de apenas semanas. Permite recolectar suficiente información para entender los requerimientos.
- Iteración del modelo funcional: Se enfoca en refinar los aspectos del sistema basados en el negocio, es decir requerimientos de información y procesamiento de alto nivel.
- Iteración de diseño y construcción: En esta fase se aplica la ingeniería en el sistema junto con los estándares, lo suficientemente altos para realizar una entrega segura al usuario.
- Implementación: Cubre la brecha entre el entorno de desarrollo y el entorno operacional.
- Post-Proyecto: Estas tareas incluyen medidas de cómo se está comportando el sistema desplegado, y si es necesario realizar mejoras. Estas mejoras suelen tener lugar 6 meses después que haya terminado la parte técnica del proyecto.

Adicionalmente, existen 9 principios que son esenciales en una implementación *DSDM*. El ignorar uno de ellos romperá la filosofía incrementando los riesgos del proyecto, a diferencia de la estructura de *DSDM*, que puede ignorar ciertos pasos. Dichos principios se mencionan a continuación:

- El involucramiento activo del usuario es imperativo
- Se debe otorgar a los equipos la potestad para tomar decisiones
- Enfocarse en entregas frecuentes



- El criterio para entregables aceptados es ajustarse al negocio
- Desarrollo iterativo e incremental es obligatorio
- Todos los cambios durante el desarrollo deben ser reversibles
- Los requerimientos son basados en alto nivel
- Las pruebas está integrado a través del ciclo de vida.
- Enfoque colaborativo y cooperativo

Programación extrema

La programación extrema (XP – Extreme Programming) es una metodología ágil, la misma que tiene como objetivo central el potenciar las relaciones interpersonales con el fin de que se el punto clave para lograr el éxito en cuanto el desarrollo de software, de esta manera se promueve un alto trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, favoreciendo un clima bastante agradable en el trabajo (Ramírez & Flórez, 2014). Es así, que la programación extrema pone más énfasis en la adaptabilidad que en la previsibilidad, a diferencia de las metodologías tradicionales. Finalmente como ventajas se puede mencionar a la programación organizada, menor tasa de errores y satisfacción del programador; mientras que las desventajas serian la recomendación de empleo para proyecto a corto plazo y altos costes en caso de fallar.

Desarrollo basado en funcionalidades

El desarrollo orientado a funcionalidades (*FDD* – Functionality Driven Development) es un método de desarrollo de software para desarrollar sistemas críticos de negocios. Está enfocado únicamente en las fases de diseño y construcción. El método recoge las prácticas reconocidas en la industria como efectivas para fundamentar el desarrollo iterativo, lo cual hace que los procesos *FDD* tengan una aplicación particular para cada caso. Además enfatiza aspectos de calidad dentro del proceso, incluyendo entregas tangibles frecuentes que permiten monitorizar de forma precisa el progreso del proyecto (P Abrahamsson et al., 2003).

Consta de cinco procesos secuenciales, y provee los métodos, técnicas y líneas guía que requieren los interesados del proyecto para desarrollar el sistema. Adicionalmente, incluye roles, artefactos, metas y líneas de tiempo necesarias en el proyecto (Pekka Abrahamsson, Salo, Ronkainen, & Warsta, 2002). Estos procesos se muestran en la Figura 2-4 y son descritos a continuación:



Figura 2-4: Metodología FDD (Pekka Abrahamsson et al., 2002).

- Desarrollo de un modelo global

Se tiene en cuenta el alcance, contexto y requerimientos del sistema a desarrollarse, utilizando casos de uso, especificaciones funcionales y modelos de objeto. Se aborda el dominio en diferentes áreas del dominio, y se hace un recorrido por cada una de ellas por los expertos del dominio junto con los desarrolladores.

- Elaborar una lista de características

Los recorridos, modelos de objeto y requerimientos existentes dan una buena base para construir una lista de características que consisten en las funciones del sistema desde la perspectiva del cliente. Esta lista es revisada por los usuarios y patrocinadores del sistema para ser validada.

- Planear por características

Consiste en la creación de un plan de alto nivel, donde los conjuntos de características se ordenan de acuerdo a una prioridad y son asignadas a los programadores. Se puede incluir una calendarización.

- Diseñar y construir por características

Se realiza el proceso iterativo para producir las características definidas en el plan. Cada iteración puede durar desde unos pocos días hasta máximo dos semanas y se incluyen tareas como inspección de diseño, codificación, pruebas unitarias, integración e inspección de código, como se muestra en la Figura 2-5.

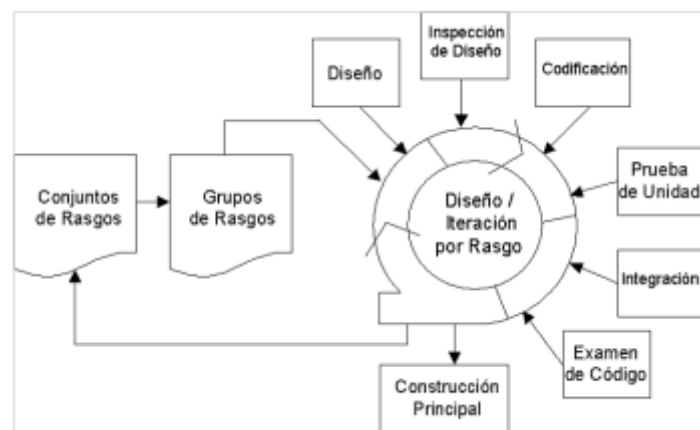


Figura 2-5: Iteraciones FDD (Pekka Abrahamsson et al., 2002).

Agile modeling

Modelado ágil (AM – Agile Modeling) es una metodología que se enfoca básicamente en realizar el modelado y la generación de documentos de manera efectiva, la cual está alineada con los principios del desarrollo ágil y puede ser utilizado por sustituto del UML estándar. AM contiene valores, principios y prácticas para el modelado de software para una aplicación de una manera efectiva y ligera.

Por otro lado logra eficacia y agilidad al momento del modelado ya que los modelos son creados de la manera más simple, fáciles de entender, lo suficientemente detallados, precisos y consistentes. Sin embargo la metodología no incluye actividades de programación y prueba, gestión de proyectos y despliegue y mantenimiento de sistemas. Por lo que AM se suele combinar con otro método para complementarlos como XP, DSDM o RUP.



En cuanto a los valores y principios de *AM* se puede decir que son similares y se basan a los de *XP*. En la siguiente lista se mencionan a las prácticas, en donde las centrales son las primeras cuatro y las siguientes tres son opcionales (Stojanovic et al., 2003):

- Modelado iterativo e incremental (aplicar los artefactos correctos, crear varios modelos en paralelo, iterar a otro artefacto, Modelo en pequeños incrementos).
- Trabajo en equipo (Modelo con otros, Participación activa de partes interesadas, Propiedad colectiva, Mostrar modelos públicamente).
- Simplicidad (Crea contenido simple, Representa modelos simplemente, Usa las herramientas más simples).
- Validación (considerar la capacidad de prueba, probarlo con el código).
- Productividad (aplique estándares de modelado, aplique patrones suavemente, reutilice los recursos existentes).
- Documentación (descartar modelos temporales, formalizar modelos de contrato, actualizar solo cuando sea estrictamente necesario).
- Motivación (Modelo para comunicarse, Modelo para comprender).

Extreme modeling

Modelado extremo (*EM* – Extreme Modeling) es un proceso de desarrollo de software que procura hacer una síntesis de procesos basados en modelos junto con programación extrema (*XP*) pero de forma diferente al modelamiento ágil. Este proceso aún se encuentra en investigación, existiendo gran cantidad de investigaciones publicadas sobre este tema. *EM* unifica los principios del modelado basado en UML y programación extrema. Para que se pueda dar esta integración se deben cumplir estos requerimientos: los modelos necesitan ser ejecutables y deben ser propensos a testing. Por lo tanto, requiere de herramientas que permitan ejecutar modelos *UML* y probarlos, soportar la transición de modelos a código y mantener el código y el modelo en sincronía (Stojanovic et al., 2003).

2.5.4 Comparativa de metodologías ágiles

A continuación en la Tabla 2-B se muestra un resumen del análisis realizado por Pekka Abrahamsson et al. (2002), por cada método usando tres aspectos: puntos clave, características especiales y deficiencias. Los puntos clave contemplan los principales aspectos o soluciones, características especiales describen uno o más aspectos de los métodos que los diferencian de los otros. Finalmente, las deficiencias encontradas se refieren a los aspectos de los cuales carece un método o que se hayan encontrado documentadas en la literatura.

Metodología	Puntos Clave	Características especiales	Deficiencias identificadas
ASD	Cultura adaptativa y colaborativa, desarrollo iterativo basado en la misión y en componentes.	Las organizaciones se consideran sistemas adaptativos. Crear un orden emergente a partir de una red de individuos interconectados.	ASD tiene más que ver con los conceptos y la cultura que con la práctica del software.



AM	Aplicación de principios ágiles al modelado: cultura ágil, organización del trabajo para apoyar la comunicación, simplicidad.	El pensamiento ágil también se aplica al modelado.	Esta es una buena filosofía complementaria para los profesionales del modelaje. Sin embargo, solo funciona dentro de otros métodos.
Crystal	Familia de métodos. Cada uno tiene los mismos valores y principios básicos subyacentes. Las técnicas, roles, herramientas y estándares varían.	Principios de diseño de métodos. Posibilidad de seleccionar el método más adecuado según el tamaño y la criticidad del proyecto	Demasiado pronto para estimar: solo existen dos de cuatro métodos sugeridos.
DSDM	Aplicación de controles a RAD, uso de timeboxing, equipos DSDM habilitados, consorcio activo para dirigir el desarrollo del método.	Primer método de desarrollo de software verdaderamente ágil, uso de creación de prototipos, varias funciones de usuario: "embajador", "visionario" y "asesor".	Si bien el método está disponible, solo los miembros del consorcio tienen acceso a documentos técnicos que tratan sobre el uso real del método.
EM	Realiza una síntesis de procesos basados en modelos junto con programación extrema.	Unifica los principios del modelado basado en UML y programación extrema.	Requiere herramientas que permitan ejecutar modelos UML y probarlos, soportar la transición de modelos a código y mantener sincronía entre código y modelo.
XP	Desarrollo dirigido por el cliente, pequeños equipos, construcciones diarias.	Refactorización: el rediseño continuo del sistema para mejorar su rendimiento y capacidad de respuesta al cambio.	Si bien las prácticas individuales son adecuadas para muchas situaciones, se presta menos atención a las prácticas generales de gestión y visualización.
FDD	Proceso de cinco pasos, basado en componentes y orientado a objetos. Iteraciones muy cortas: de horas a 2 semanas.	Simplicidad de métodos, diseño e implementación del sistema por características, modelado de objetos.	FDD se enfoca solo en el diseño y la implementación. Necesita otros enfoques de apoyo.
Scrum	Equipos de desarrollo independientes, pequeños y auto organizados, ciclos de	Haga cumplir un cambio de paradigma de "definido y repetible" a la "nueva vista de	Si bien Scrum detalla específicamente cómo administrar el ciclo de publicación de 30 días, las pruebas de



	lanzamiento de 30 días.	desarrollo de productos de Scrum".	integración y aceptación no son detalladas.
--	-------------------------	------------------------------------	---

Tabla 2-B: Comparativa metodologías ágiles (Pekka Abrahamsson et al., 2002).

2.6I-Estrella (i*)

I* es un marco que se desarrolló para obtener los requerimientos en una fase temprana, con el fin de modelar y razonar sobre entornos organizacionales y sus sistemas de información. Los modelos i* permiten la exploración del dominio del sistema haciendo énfasis en aspectos sociales, representándolos de forma gráfica mediante actores del sistema, incluyendo sus intenciones, dependencias y alternativas (Yu, 1997). El enfoque orientado hacia agentes de i* está representado por actores (agentes y roles), y las asociaciones entre ellos.

Los actores dependen unos de otros para completar tareas, utilizando recursos, mediante objetivos y objetivos blandos. Los objetivos blandos hacen referencia a objetivos sin criterios claros a ser satisfechos; por lo tanto, se logra un objetivo blando cuando se considere que éste está suficientemente satisfecho. Las intenciones que motivan las dependencias son exploradas para cada actor, teniendo en cuenta los objetivos, objetivos blandos, tareas y recursos deseados por dichos actores (Horkoff & Yu, 2016).

Las interrelaciones entre las intenciones dentro de un actor se representan a través de tres tipos de enlaces. (i) *Enlaces de Descomposición*: muestran las intenciones que son necesarias para lograr una tarea. (ii) *Enlaces Medio-Fin*: muestran las tareas alternativas que pueden lograr un objetivo. (iii) *Enlaces de Contribución*: muestran los efectos de objetivos blandos, objetivos y tareas en objetivos blando.

Las contribuciones positivas o negativas que permiten satisfacer o negar un objetivo blando son representadas por enlaces *Make* (contribución positiva) o *Break* (contribución negativa), respectivamente. Las contribuciones que evidencian ser positivas o negativas de forma parcial, que no son en sí mismas suficiente para satisfacer o negar un objetivo blando, están representados por enlaces *Help* o *Hurt* (Horkoff & Yu, 2016).

2.7 Sistemas de información web

También se llaman sistemas de información basados en la web, son sistemas que brindan información y servicios a usuarios o a otros sistemas mediante tecnologías web de internet. Por lo general se conforma por una o más aplicaciones web, componentes específicos orientados a servicios, junto con componentes de información. Liang (2006) propone una estructura general en donde un WIS consta de dos partes: una parte del sitio web (*WS* – Web Site), en donde se muestran servicios e información al usuario, y una segunda parte de sistemas de información (*IS* – Information System), que los proporciona. Así mismo, un WIS trabaja como un IS, integrando varios componentes como base de datos y sistemas transaccionales, mediante el uso de diferentes tecnologías web como navegadores, protocolos y estándares que apoyan sus funciones básicas.

2.8 Meta modelo de ingeniería de procesos de software

(*SPEM* - Software Process Engineering Meta Model) es un lenguaje de modelado especificado por la (*OMG* - Object Management Group) a manera de estándar que considera



un alto nivel de abstracción y que es usado para describir los procesos que se realizan para el desarrollo de software orientado a objetos. SPEM se basa en (*MOF* - Meta Object Facility), siendo un meta modelo (*UML* - Uniform Model Language) (OMG, 2008). De esta manera representa a los procesos de desarrollo de software y sus componentes relacionando a dichos procesos con una colaboración entre entidades activas abstractas denominadas Roles, que son los que realizan las actividades en entidades concretas, tangibles denominadas productos de trabajo (Menéndez Domínguez & Castellanos Bolaños, 2008).

2.9 Modelo y notación de procesos de negocio

Los modelos y notaciones de procesos de negocios (*BPMN* – Business Process Model and Notation) comprenden un estándar propuesto por la *OMG* que permite representar de forma gráfica y expresiva los procesos de negocio, siguiendo un formato de flujo de trabajo. El objetivo principal de *BPMN* es proporcionar a los interesados o involucrados del negocio una notación gráfica estándar, que tenga las características de ser entendible y legible con facilidad. De esta manera, se le considera un lenguaje común para permitir una mejor comprensión de los que se realiza en las organizaciones (Chinosi & Trombetta, 2012).

2.10 Certificados X.509

Los certificados X.509 son certificados digitales que mediante una infraestructura de clave pública X.509, permiten encontrar la correspondencia entre una clave pública y su respectiva entidad, dentro de un certificado. La generación de estos certificados la realiza una entidad de confianza, la cual se conoce como una Autoridad de Certificación (*CA* – Certification Authority).

Los beneficios encontrados en los certificados, frente a otros mecanismos de seguridad, están dados por la identificación y autenticación, aprovechando las utilidades de claves asimétricas con los dispositivos. De esta manera, se almacenan claves privadas de forma interna en cada dispositivo, protegiendo este material criptográfico de forma confidencial (Amazon, 2018).

2.11 Autoridad de certificación

Se conoce a la autoridad de certificación (*CA* – Certification Authority) como una entidad cuyas funciones son garantizar autenticidad y veracidad de los datos incluidos en un certificado digital por analizar. De esta manera, la *CA* realiza tareas notariales, con lo cual ofrece fidelidad jurídica. Por medio de una clave privada se puede identificar al propietario de un certificado digital, al realizar la firma electrónica de certificados emitidos. Como norma la autoridad de certificación debe disponer de un servicio para validación de certificados, dado que estos pueden establecerse como revocados, es decir, perder su validez por caducidad o sanción.

La emisión de certificados se realiza hacia personas, empresas o autoridades de certificación de menor rango, con lo cual se establece una jerarquía para emisión de certificados. Esta estructura se conforma basándose en una autoridad de certificación auto-firmada, la cual se establece como raíz de la jerarquía. De esta manera, el certificado de la última *CA* es avalado, con lo cual se asegura su autenticidad cuando se lo instala en el almacén de certificados del computador. De esta forma se hace posible descargar el



certificado principal de la CA desde el sitio web y generando confiabilidad en la propia página (E-administracion, 2018).

2.12 Firma electrónica

Una firma electrónica es un conjunto de datos electrónicos, los cuales se asocian a un documento digital, permitiendo identificar al firmante y validar la integridad de dicho documento, asegurando el no repudio de este documento firmado. Para firmar un documento electrónico se requiere un certificado digital, el cual contiene claves criptográficas necesarias para realizar la firma (Portal de Administración Electrónica, 2018).

En general para firmar un documento electrónicamente, se siguen los siguientes pasos:

- Disponer del documento a firmar y de un certificado digital.
- La aplicación realiza un resumen del documento. El resumen generado es único, siendo alterado inmediatamente con cualquier modificación del documento.
- La aplicación usa la clave del certificado para realizar una codificación del resumen.
- Se crea un nuevo documento que contiene este resumen codificado. Esto será la firma electrónica.

2.13 Tecnologías involucradas

1. Marco de desarrollo de modelado Eclipse

El marco de desarrollo de modelado Eclipse (EMF – Eclipse Modeling Framework) es un marco de modelado y un recurso que facilita la generación de código, con el fin de construir herramientas y otras aplicaciones basadas en modelos de datos estructurados. A partir de una especificación de modelo descrita en *XMI*, *EMF* proporciona herramientas y soporte de tiempo de ejecución para producir un conjunto de clases de Java para el modelo, junto con un conjunto de clases de adaptadores que permiten la visualización y edición basada en comando del modelo y un editor básico (Berlik, 2007).

2. Servicios web

Según Booth et al. (2004) un servicio web es “un sistema de software diseñado para admitir la interacción interoperable de máquina a máquina a través de una red. Tiene una interfaz descrita en un formato procesable por máquina (especificación WSDL)”. Otros sistemas interactúan con el servicio web de la manera prescrita por su descripción utilizando mensajes SOAP, normalmente transmitidos mediante HTTP con una serialización XML junto con otros estándares relacionados con la Web.

3. Computación en la nube

El Instituto Nacional de Estándares y tecnología (*NIST*) define la computación en la nube (*cloud computing*) como “un modelo para permitir el acceso de la red ubicuo, conveniente y bajo demanda a un grupo compartido de recursos informáticos configurables (ej. Redes, servidores, almacenamiento, aplicaciones y servicios) que pueden aprovisionarse y liberarse



rápidamente con un mínimo esfuerzo de gestión o una intervención del proveedor de servicios” (Mell & Grance, 2011).

4. Glassfish

Es un servidor de aplicaciones desarrollado por Sun Microsystems, la misma que pertenece a Oracle Corporation, dispone de una licencia libre. Implementa las especificaciones de Java Platform Enterprise Edition (Java EE), la cual brinda flexibilidad (Oracle, 2005).

5. Java

Es un lenguaje de programación orientado a objetos para desarrollar cualquier tipo de sistemas. Es uno de los más utilizados hoy en día, ya que tiene varias ventajas y beneficios para los programadores. Entre las más relevantes tenemos que es un lenguaje independiente de la plataforma y adaptable a todo tipo de dispositivo móvil, entre otro (Silvestrini, 1985).

6. MySQL

Es un sistema de gestión de base de datos relacional considerado el más popular del mundo, dispone de una licencia de tipo pública y comercial por Oracle Corporation. Dispone de confiabilidad, uso sencillo y rendimiento de altas prestaciones. MySQL se ha convertido en la principal opción de base de datos para aplicaciones basadas en la Web (Oracle, 2018).

7. Protocolo de acceso simple a objetos

El protocolo de acceso simple a objetos (*SOAP* – Simple Object Access Protocol) es un protocolo estándar que realiza las definiciones para el intercambio de información estructurada en un entorno descentralizado y distribuido. Su intercambio de datos se maneja mediante la tecnología *XML*. Es uno de los protocolos utilizados en los servicios web, el mismo que ha sido diseñado para ser independiente de cualquier modelo de programación particular y otra semántica específica de la implementación (W3C, 2007).

8. Protocolo de transferencia de hipertexto

El Protocolo de transferencia de hipertexto (*HTTP* – Hypertext Transfer Protocol) es un protocolo a nivel de aplicaciones para sistemas de información hipermedia distribuidos, colaborativos. Es un protocolo genérico de comunicación que permite la transferencia de información en la web. Adicionalmente es un protocolo sin estado, es decir, no guarda ninguna información sobre conexiones anteriores. Además de su uso para hipertexto puede ser utilizado como servidores de nombres y sistemas de administración de objetos distribuidos, mediante la extensión de sus métodos de solicitud, códigos de error y encabezados. Otra característica de *HTTP* es el tipado y la negociación de la representación de datos, permitiendo que los sistemas se construyen independientemente de los datos que se transfieren (W3C, 1999).

9. POST

POST es un método del protocolo *HTTP* que permite el envío de datos al servidor. Generalmente es enviado por un formulario (*HTML* – Hyper Text Markup Language),



generando acciones en el servidor. La solicitud contiene un cuerpo y su tipo es definido por la cabecera Content-Type (Mozilla, 2017).

10. Protocolo simple de transferencia de correo

El protocolo simple de transferencia de correo (*SMTP* – Simple Mail Transfer Protocol) es un protocolo de red, *TCP/IP* que es usado para el intercambio de correos electrónicos, envío y recepción de estos. Como se sabe los protocolos son un conjunto de normativas y reglas que posibilitan la circulación de información en la red. Adicional *SMTP* es utilizado con los protocolos *POP3* o *IMAP* para el almacenamiento de los correos en un buzón del servidor y bajarlos periódicamente para el usuario (Klensin, 2008).

11. Servicio de mensajes cortos

El servicio de mensajes cortos (*SMS* – Short Message Service) es un servicio que se encuentra disponible para los teléfonos móviles, el cual originalmente fue diseñado como parte del estándar (*GSM - Global System for Mobile communications*) de telefonía móvil digital. También es conocido como mensajes de texto, el mismo que permite el envío de mensajes cortos (Condes, 2005).

12. Correo electrónico

El correo electrónico es un servicio que permite el manejo de mensajes entre usuarios, mediante redes de comunicación electrónica, permitiendo el envío y recepción de los mismos. Mediante este servicio es posible no solo el envío de texto, además es posible adjuntar archivos digitales, con una exigencia en cuanto al límite de su tamaño (Computerhope, 2018).

13. Amazon Web Services

AWS por sus siglas en inglés, es una plataforma que brinda servicios en la nube, infraestructura de *TI*, ofreciendo características como potencia de cómputo, almacenamiento de bases de datos, entrega de contenido, entre otras. Brinda un servicio bajo demanda, es decir, se paga solo por lo que se utiliza (Amazon, 2016)

Capítulo 3

Estado del arte

El presente capítulo realiza una revisión sistemática de la literatura apegado al enfoque de los objetivos de esta investigación. Tiene como objetivos la búsqueda, identificación y análisis de estudios relacionados al dominio. De esta manera, se puede conocer el estado actual de la temática, así como hacia donde están orientadas las diferentes investigaciones y las tecnologías utilizadas.

Las revisiones sistemáticas han sido muy comunes en el área de la Ingeniería del Software, y sirven para identificar las brechas de investigación. Si bien se ha buscado en la literatura por estudios secundarios que aborden este tema, no ha sido posible encontrar uno que responda las preguntas e inquietudes relacionadas a este trabajo de titulación.

En este contexto, este capítulo muestra una revisión sistemática en la que se busca generar conocimiento y conocer el estado de la investigación en temas relacionados a facturación electrónica, aspectos tecnológicos relacionados y sistemas de notificación. De esta manera, se presenta un estudio secundario que nos ha permitido resolver preguntas de investigación preliminares, relacionadas al establecimiento de un método para desarrollo de sistemas de información de notificaciones para la gestión de cobranzas en organizaciones, que permitan el envío de mensajes a sus clientes a través de medios digitales, ajustados a la normativa ecuatoriana en la gestión de comprobantes electrónicos.

3.1 Método de estudio

En esta sección se presenta el estado actual del tema de titulación, para que de esta manera quede totalmente clarificado de una manera formal. Se ha visto pertinente realizar una revisión sistemática basada en la metodología de Kitchenham y Charters (2007). Esta metodología propone una guía para realizar un estudio sistemático, el cual incluye un conjunto de actividades específicas para su ejecución. Dichas actividades se dividen en tres fases: i) Planeación, ii) Ejecución y iii) Reporte. De esta manera se obtiene un resumen en cuanto a la evidencia, beneficios y desventajas de la temática, así como identificar aquellos nichos en la investigación actual, que permitan una investigación mayor en estos temas.

3.1.1 Etapa de planificación

Esta fase consiste en establecer los pasos que se realizarán durante la revisión, para esto se determinan preguntas de investigación. Por otro lado, se definen estrategias de búsqueda y evaluación de la calidad para evitar amenazas de validez. Dicha etapa se compone de dos pasos: (i) establecer las preguntas de investigación, (ii) definir las fuentes de datos y la estrategia de búsqueda.

3.1.1.1 Establecer las preguntas de investigación

La pregunta de investigación que se ha planteado define el alcance de la revisión sistemática y es la siguiente: “¿Cuál es el estado del arte respecto a facturación electrónica y plataformas o sistemas de información que utilizan notificaciones?”

Sub-preguntas de investigación

Para abordar la pregunta de investigación se han propuesto las siguientes sub-preguntas:

- RQ1: ¿Cómo se maneja el proceso de facturación electrónica en otros países?
- RQ2: ¿De qué manera se aborda el aspecto técnico en las implementaciones de facturación electrónica?
- RQ3: ¿Qué atributos de seguridad se consideran en el proceso de facturación electrónica?
- RQ4: ¿Cómo se están abordando los estudios de facturación electrónica y de qué manera los validan?

3.1.1.2 Fuente de datos y estrategia de búsqueda

Se han seleccionado las siguientes librerías digitales para la búsqueda automática: IEEE Xplore, ACM Digital Library, Springer Link, Science Direct, Scholar Google. Para la búsqueda manual se han elegido las mejores conferencias y revistas relacionadas al tópico de comercio electrónico.

Búsqueda manual

En la Tabla 3-A se muestran las conferencias y revistas seleccionadas donde se han encontrado artículos que no pudieron ser recuperados por la búsqueda automática y cuyo contenido es representativo para la investigación.

Conferencias
IEEE International Conference on Enterprise Computing, E-Commerce and E-Services
International Conference on Electronic Commerce
International Symposium on Data, Privacy and E-Commerce
Revistas
International Journal of Electronic Commerce
Journal of Organizational Computing and Electronic Commerce

Tabla 3-A: Fuentes de búsqueda manual.

3.1.1.3 Cadena de búsqueda

Para realizar la búsqueda automática se definió una cadena basada en palabras clave relacionadas con el tema de la investigación y elegida según los criterios de los investigadores, luego de un proceso de múltiples intentos con diferentes combinaciones de posibles palabras clave. Se ha utilizado el inglés para dichas búsquedas, por ser el idioma reconocido mundialmente y utilizado como estándar en el campo de la investigación. Estas combinaciones y la cadena de búsqueda final se muestran en la Tabla 3-B.

Concepto	Subcadena	Conector	Términos alternativos
Facturación electrónica	e?invoic*	OR	incluye e-invoice, e-invoicing
Facturación electrónica	electronic invoic*	OR	incluye electronic



			invoice, electronic invoicing
Facturación electrónica	electronic billing	AND	
Plataforma	platform	OR	
Sistemas de información	information system	OR	
Sistemas	system	OR	
Web	web	OR	
Computación en la nube	cloud	AND	
Notificaciones	notification	OR	
Alertas	alert	OR	
Mensajes	message	OR	
Mensajes	SMS	OR	
Correo electrónico	mail		
String de búsqueda	("e-invoice" OR "electronic invoice" OR "electronic billing") AND ("platform" OR "information system" OR "web" OR "cloud") AND ("notification" OR "alert" OR "message" OR "SMS" OR "mail")		

Tabla 3-B: Cadena de búsqueda.

3.1.1.4 Período de búsqueda

La facturación electrónica tiene sus inicios en los años 1970 cuando las organizaciones empezaron a intercambiar archivos de facturas. En ese entonces no existían estándares para el intercambio y los sistemas de negocio no se ajustaban a las exigencias que se requerían para ese propósito. Soluciones tecnológicas y estándares como *XML* hicieron posible este proceso (Sundström, 2006).

3.1.1.5 Criterios de extracción

Para responder las preguntas de investigación planteadas, se han definido criterios de extracción de información, los cuales se muestran en la Tabla 3-C. La importancia de definir los criterios de extracción radica en evitar el sesgo de los investigadores para que no influyan en sus expectativas en los análisis de los estudios.

RQ1: ¿Cómo se maneja el proceso de facturación electrónica en otros países?		
EC1	Modelo de Servicio	Infraestructura como servicio (<i>IaaS</i> - Infrastructure As A Service), Plataforma como servicio (<i>PaaS</i> - Platform As A Service), Software como servicio (<i>SaaS</i> - Software As A Service)
EC2	Modelo de Despliegue	Comunitario, Privado, Público
EC3	Tipos de institución	Comerciales, Gubernamentales, Industriales, Educación, Otras
EC4	Partes que incluye	Notificaciones, Facturación electrónica,



		Computación en la nube, Tecnologías
EC5	Usos	Pago de mercancía, Pago de servicios, Pago de impuestos
RQ2: ¿De qué manera se aborda el aspecto técnico en las implementaciones de facturación electrónica?		
EC6	Tipos de arquitectura	Monolítica, Orientada a servicios, Microservicios
EC7	Metodología de desarrollo utilizada	Scrum, Cascada
EC8	Lenguaje de programación (Back End)	Java, PHP, .NET
EC9	Lenguajes de programación (Front End)	Javascript,HTML,Angular, Bootstrap
EC10	Base de datos	Relacionales, No relacionales
RQ3: ¿Qué atributos de seguridad se consideran en el proceso de facturación electrónica?		
EC12	Atributos de seguridad	Integridad, Confidencialidad, No repudio, Autenticidad
RQ4: ¿Cómo se están abordando los estudios de facturación electrónica y de qué manera los validan?		
EC13	Cómo se abordan los estudios	Industria, Academia, Caso de estudio
EC14	Métodos de validación	Experimento controlado, Pruebas de conceptos
EC15	Tipo de estudio	Nuevo, Extensión

Tabla 3-C: Criterios de extracción.

3.1.2 Etapa de ejecución

Esta fase consiste en establecer las pautas para seleccionar los estudios de una manera adecuada, así mismo evaluarlos por su calidad. De esta manera esta etapa consiste de tres pasos: (i) selección de estudios primarios, (ii) evaluación de calidad y (iii) método de análisis y síntesis.

3.1.2.1 Selección de estudios primarios

Luego de realizar la búsqueda automática y manual los autores proceden a evaluar cada estudio para determinar si debería o no ser incluido. Cualquier discrepancia en este proceso se resuelve en consenso al examinar completamente el estudio en conflicto.

Serán incluidos los estudios que cumplan por lo menos uno de los siguientes criterios de inclusión:

- Estudios que contienen información sobre facturación electrónica
- Estudios que involucran a facturación electrónica y computación en la nube.
- Estudios que contienen información sobre notificaciones electrónicas
- Estudios que contienen información relacionada a la calidad y tecnologías que utilizan dichas plataformas.

Serán excluidos los estudios que cumplan alguno de los siguientes criterios de exclusión:

- Artículos introductorios de temáticas especiales.
- Estudios duplicados del mismo tema en diferentes fuentes.
- Short artículos de menos de cinco páginas.
- Artículos escritos en idiomas diferentes al castellano y al inglés.

Esta etapa describe los resultados de las tareas realizadas en este estudio. La búsqueda automática de la cadena de búsqueda mencionada anteriormente permitió recuperar estudios de distintas fuentes o bibliotecas digitales, considerando estudios incluidos y excluidos.

Después de aplicar los criterios de inclusión y exclusión en los estudios seleccionados de las búsquedas, se consideraron 39 artículos, como se muestra en la Tabla 3-D. Por otro lado, se consideraron 9 artículos para este estudio en la búsqueda manual.

	Incluido	No incluido	Porcentaje
ACM	3	71	7.69 %
IEEE XPLORE	11	162	28.21 %
SCHOLAR GOOGLE	20	598	51.28 %
SPRINGER LINK	2	50	5.13 %
SCIENCE DIRECT	3	282	7.69 %
TOTAL	39	1163	100.00 %

Tabla 3-D: Número de artículos incluidos.

3.1.2.2 Evaluación de la calidad

Además de los criterios de inclusión y exclusión, es importante llevar a cabo una evaluación de calidad. Para este propósito, se usa una escala de Likert de tres puntos, basada en el número de citas de cada estudio. La Tabla 3-E muestra los resultados de la evaluación de acuerdo con los criterios de calificación: (-1) el estudio no tiene citas, (0) el estudio tiene de una a tres citas, (+1) el estudio tiene más de tres citas.

Puntaje	Cantidad	Porcentaje
0	6	12.50 %
+1	34	70.83 %
-1	8	16.67 %
TOTAL	48	100 %

Tabla 3-E: Evaluación de calidad.

3.1.2.3 Método de análisis y síntesis

En esta sección presentamos los resultados finales de la revisión sistemática, los histogramas muestran los resultados finales de cada uno de los criterios que se ha

seleccionado de mayor relevancia, por otro lado se obtuvieron diferentes gráficos de burbujas. Finalmente se plasma una discusión sobre dichos temas.

La Figura 3-1 muestra el porcentaje de estudios según el tipo de institución a la que están enfocados, lo que demuestra que existe un gran interés en el sector comercial y gubernamental. Por otro lado, la Figura 3-2 muestra un porcentaje de distribución de los estudios según el tema tratado, encontrando la mayoría en temas de facturación electrónica y una menor cantidad en cuanto a la parte de notificaciones.

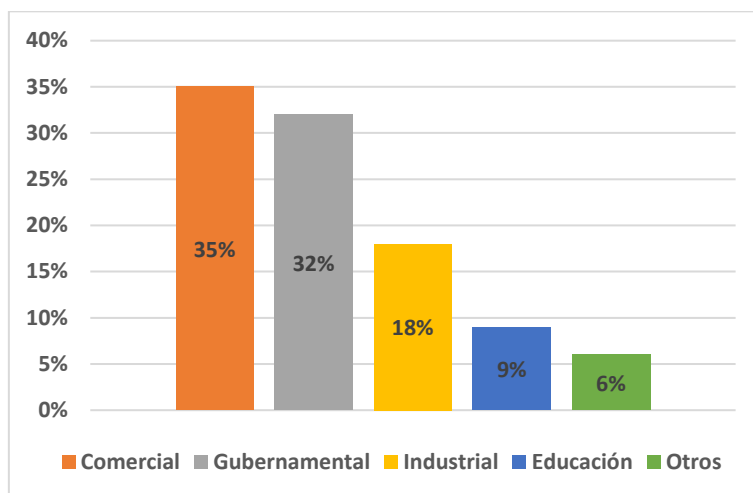


Figura 3-1: Porcentaje de estudios correspondiente a EC3: Tipos de instituciones.

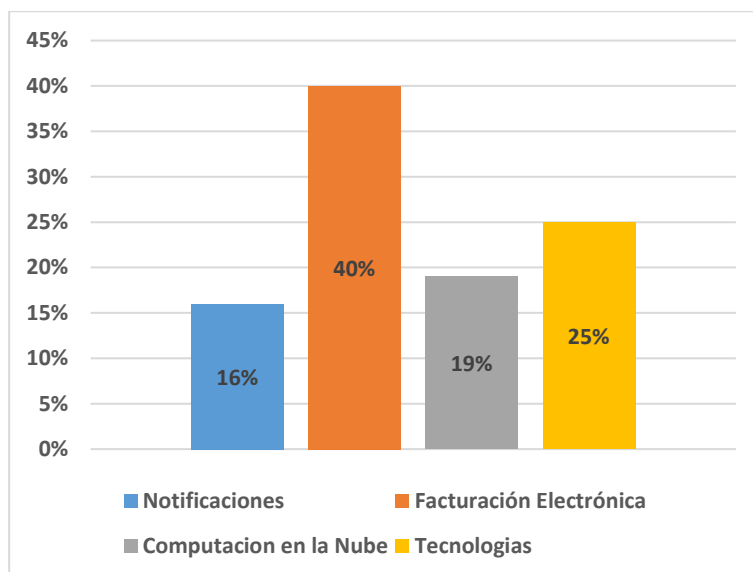


Figura 3-2: Porcentaje de estudios correspondientes a EC4: Partes que incluyen.

La Figura 3-3 muestra el porcentaje de estudios que aborda aspectos de seguridad, mostrando un principal interés en la autenticidad, seguido de valores cercanos entre integridad, confidencialidad y no repudio. Finalmente, la Figura 3-4 muestra el porcentaje de cómo se están abordando los estudios analizados, mostrando un enfoque académico en primer lugar, seguido de casos de estudio y finalmente industrial.

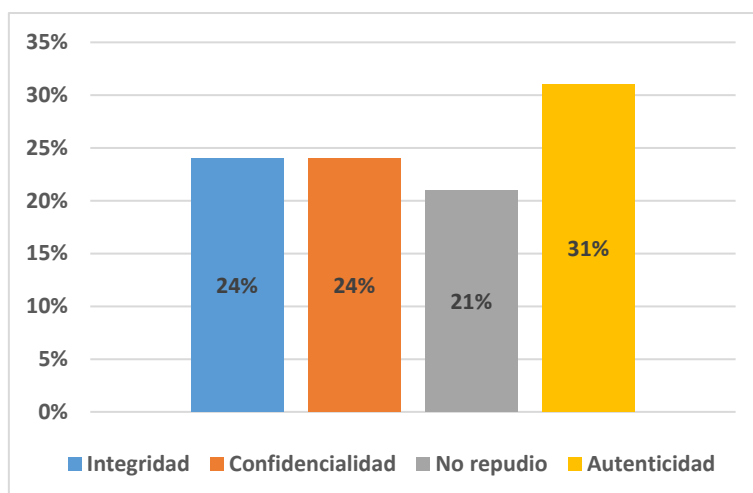


Figura 3-3: Porcentaje de estudios correspondientes a EC11: Atributos de seguridad.

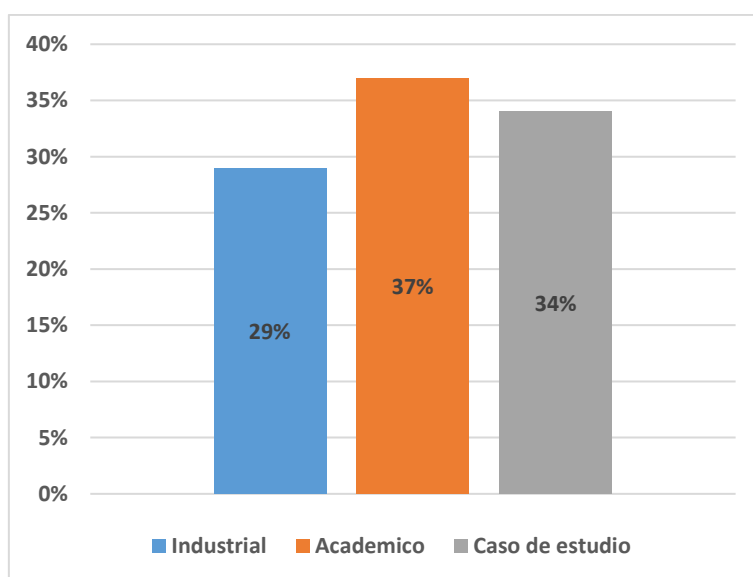


Figura 3-4: Porcentaje de estudios correspondientes a EC12: Como se abordan los estudios.

Por otro lado, la Figura 3-5 contiene el eje de las ordenadas representadas por EC 3: tipos de institución, en el eje de las abscisas se muestra EC 4: partes incluidas. De esta manera, se observa que la mayoría de los estudios se están llevando a cabo sobre temas de facturación electrónica centradas en instituciones comerciales, seguidas por el gobierno. También la computación en la nube es un punto de interés en las instituciones gubernamentales, mientras que el aspecto de las notificaciones apenas se está abordando. De forma similar, la Figura 3-6 muestra en el eje de abscisas a EC11: atributos de seguridad, en el eje de ordenadas representa el EC 3: tipos de institución y partes incluidas. De esta manera, se observa que en muchos estudios relacionados con la facturación electrónica se abordan los atributos de seguridad, mientras que en los estudios sobre notificaciones no se aborda el tema de la seguridad. Además, los atributos de seguridad son un punto de interés principalmente en estudios que tienen un enfoque comercial.



3.1.3 Etapa de reporte de la revisión

A continuación hay una breve discusión de acuerdo con los criterios de extracción que se han considerado:

EC3: Tipos de institución

La facturación electrónica se ha considerado como una de las fuentes más importantes de rentabilidad en Europa (Penttinen & Hyytiäinen, 2008). Algunos países han sido más activos en la transición a la facturación electrónica. En algunos casos, esto ha sido promovido por el gobierno en el nivel de la legislación, por lo que es obligatorio, como el caso de Dinamarca, analizado por Penttinen y Hyytiäinen (2008) y el caso de Italia con las instituciones públicas.

La inclusión de la facturación electrónica en las instituciones produce una disminución en los costos de facturación y, por lo tanto, un aumento en los recursos disponibles, que podrían utilizarse para otros fines para aumentar el valor de la organización (Pintarić, Panjkota, & Zdrilić, 2010). Hay factores que influyen en la adopción de la facturación electrónica en las instituciones. El estudio de Pintarić et al. (2010) analiza estos factores con un estudio de caso de cuatro empresas privadas y dos organizaciones públicas. A continuación se muestran los factores junto con los resultados después de la implementación de la facturación electrónica:

(i) Beneficio relativo: grado en el que una innovación es mejor que la anterior. Se redujo el número de errores en las facturas en papel. El tiempo de circulación de las facturas se redujo de una semana a dos días.

(ii) Compatibilidad: grado en que una innovación es consistente con lo que ya existe. La usabilidad de los sistemas de facturación electrónica es importante, ya que sus interfaces gráficas deberían ser fáciles y similares a las facturas en papel.

(iii) Complejidad: grado en que una innovación es percibida como difícil de entender y usar. La transición a la facturación electrónica debería ser simple y tener la capacidad de centralizar las facturas de múltiples departamentos dentro de la organización.

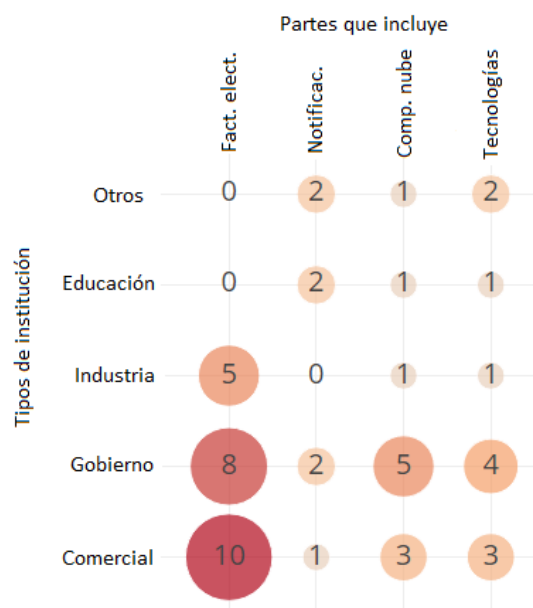


Figura 3-5: Comparación entre EC3: Tipo de instituciones y EC4: Partes que incluye.

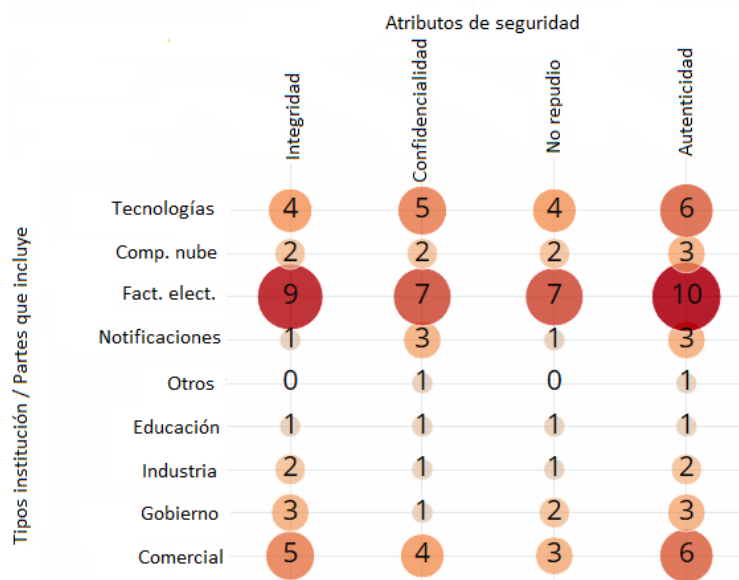


Figura 3-6: Comparación entre EC11: Atributos de seguridad, EC3: Tipos de institución y EC4: Partes que incluyen.

(iv) Capacidad de pruebas: el grado en que se puede probar una innovación. Es útil probar el funcionamiento de la facturación electrónica tomando una compañía piloto para realizar las pruebas y obtener retroalimentación.

(v) Observabilidad: grado en que los resultados de una innovación son visibles. Una organización pública busca proyectar una imagen positiva y moderna del servicio público, además de monitorear el flujo interno de información entre sus empleados.

Por otro lado, los estudios propuestos por Suwisuthikasem y Tangsripairoj (2008); Chu, Chai, Liu, y Sun (2014); Matus et al. (2017) abordan la relación de la facturación electrónica con la recaudación de impuestos, dada la validez legal de que este mecanismo proporciona.



En Suwisuthikasem y Tangsripairoj (2008) se propone una solución de facturación electrónica que permite la integración de los contribuyentes con la plataforma electrónica del gobierno que tiene acceso público. Los resultados muestran que las necesidades de los contribuyentes están resueltas y que se permite un control más eficiente de la agencia de recaudación de impuestos.

EC4: Partes incluidas

Este criterio es uno de los más importantes para la clasificación de los trabajos, ya que nos segmenta entre aspectos como notificaciones, facturación electrónica, computación en la nube y tecnologías. En términos de notificaciones, se encontraron 12 documentos que abordan este tema, donde C. H. Chiu, Wu, Tut, Lin, y Yuan (2007); Ghazal, Ali, Al Halabi, Ali, y Al Khalil (2016); Ji, Ganchev, Odroma, y Zhao (2014) abordan temas relacionados que pueden contribuir al desarrollo de sistemas de notificación de emergencias en tiempo real. Estos desarrollos proponen un sistema en el que se pueden realizar notificaciones y alertas desde una aplicación móvil, un reloj inteligente, así como *SMS* y el uso del protocolo extensible de mensajería y presencia (*XMPP*). Por otro lado Astudillo y Zambrano (2006) proponen un módulo de notificación para una escuela, donde facilita la interacción alumno-docente para el intercambio de archivos y el envío de *SMS*, que también incluye las tecnologías utilizadas para dicha implementación.

En otro artículo, se propone el uso de (IM - Instant Messenger) para notificaciones en tiempo real, como es el caso de Kiatruangkrai, Phusayangkul, Viniyakul, Prompoon y Kanongchaiyos (2010), Carpio y Faicán (2009), donde se propone una arquitectura para el uso de notificaciones a empresas con servicios web que hacen uso del formato *XML* y un sistema de gestión de contenido para el comercio electrónico, desarrollado con *HTML*, *CSS*, *PHP* y *Ajax*, respectivamente.

Por ultimo en Macario y Srirama (2013), se explora las posibilidades de implementar un marco de mensajería que se puede usar como un mecanismo de notificación gratuito que se ejecuta en la nube, el marco se basa en la mensajería extensible y el protocolo de presencia (*XMPP*). Complementando el estudio con varios mecanismos proporcionados por los proveedores de la nube para entregar mensajes a las aplicaciones móviles, estos mecanismos se consideran servicios de caja negra (por ejemplo, *GCM*, *APNS*, *MPNS*, entre otros.).

Asimismo, se encontraron 27 documentos que abordan la facturación electrónica, por un lado, menciona sus beneficios y, por otro lado, la implementación de dicho sistema. En Lian (2015), se realiza un estudio sobre los factores implicados en el gobierno electrónico haciendo uso de la computación en la nube, donde concluye como una tendencia que debe adoptarse en las empresas para obtener mayores beneficios. Del mismo modo Penttinen y Hyytiäinen (2008); Pintarić et al. (2010); Andreeff et al. (2001); Rofhök-björni (2006); Voutilainen y Pento (2003); Kim y Rohmer (2012); Kreuzer, Eckhardt, Bernius y Krönung (2013); Legner y Wende (2006); Yen y Ng (2003); Lagzian y Naderi (2014); Assimakopoulos, Riggas y Kotsimpos (2015) hablan sobre los beneficios e impactos de la adopción de sistemas o plataformas para facturación electrónica, comercio electrónico y gobierno electrónico.

Por un lado, Chu et al. (2014); Kaliontzoglou, Boutsy y Polemi (2006), Cuylen, Kosch y Breitner (2016); (Yen & Ng, 2003) proponen marcos y modelos maduros para el proceso de facturación electrónica y el sistema impositivo, a fin de tener una eficiencia para ese proceso



y mejorar la recaudación de impuestos. Por otro lado, Suwisuthikasem y Tangsripairoj (2008), Matus et al. (2017); Spanic, Ristic y Vrdoljak (2011), Netter y Pernul (2009), Vanjak, Mornar y Magdalenic (2008) proponen sistemas de facturación electrónica con diferentes funcionalidades como envío, recepción y procesamiento de facturas sin intervención manual, elaboración de patrones de seguridad para el proceso de facturación electrónica, sistemas de facturación electrónica para la recaudación de impuestos utilizando servicios web, además Matus et al. (2017) se destaca al proponer una plataforma de facturación electrónica para la recaudación de impuestos basada en computación en la nube con Software as a Service, en donde se muestra las tecnologías empleadas para el desarrollo, donde recibe, almacena, procesa y envía facturas a los recaudadores de impuestos a través del correo electrónico y también al consumidor.

EC11: Atributos de seguridad

En los sistemas de notificaciones es muy importante la seguridad en los aspectos de autenticidad, autorización y contabilización. En el sistema de notificación propuesto por C. H. Chiu et al. (2007), se utiliza el protocolo extensible de mensajería y presencia (*XMPP* - Extensible Messaging and Presence Protocol) para proveer seguridad en las comunicaciones entre las empresas y proveedores de servicio de notificaciones de tipo mensaje instantáneo (*IM* - Instant Message). Adicionalmente, la información transmitida a través de Internet mantiene la confidencialidad al utilizar comunicaciones encriptadas con el protocolo seguro de transferencia de hiper-texto (*HTTPS* - Hypertext Transfer Protocol Secure) y la utilización de firma digital con una infraestructura de clave pública (*PKI* - Public Key Infrastructure).

Como una medida de protección contra una base de datos que ha sido comprometida, vulnerada, el sistema propuesto en Kiatruangkrai et al. (2010) utiliza un algoritmo de hash MD5 para cifrar las contraseñas de los usuarios de su sistema antes de guardarlas en la base de datos. De esta manera, si un atacante obtiene acceso a la base de datos, esta información sensible se encontrara ilegible.

Dada la naturaleza del comercio electrónico entre clientes y bancos, el estudio de Rofhök-björni (2006) utiliza el protocolo de red privada virtual (*VPN* - Virtual Private Network) para verificar el origen de las comunicaciones, para garantizar la integridad y confidencialidad de los mensajes transmitidos.

El estudio de Kaliontzoglou et al. (2006) propone un sistema de facturación electrónica basado en servicios web, que cumple con los requisitos de la directiva europea 2001/115/CE. El sistema proporciona autenticación de origen con la aplicación de firmas digitales en combinación con tarjetas inteligentes. La integridad de la información se logra mediante el uso de funciones criptográficas hash, junto a la firma o de forma separada. El no-repudio de origen y destino se consigue con las firmas digitales de firma electrónica avanzada XML (*XAdES* - XML Advanced Electronic Signatures) y marcas de tiempo. La sintaxis de cifrado XML se alcanza siguiendo la Recomendación W3C. Los estudios de Spanic et al. (2011); Chu et al. (2014); Vanjak et al. (2008); Brun y Lanng (2006); Tanna, Gupta, Rao y Upadhyaya (2005) consideran los mismos mecanismos de autenticación, no repudio, integridad y confidencialidad.

EC12: Cómo se está abordando el estudio y EC13: Método de validación.

En los estudios se han encontrado 14 artículos que realizan casos de estudio como Ghazal et al. (2016); Astudillo y Zambrano (2006); Lian (2015); Voutilainen y Pento (2003);



Potapenko (2010); Alippi, Pessina y Roveri (2005), entre los más destacados se puede mencionar a Penttinen, Hallikainen y Salomäki (2009) el cual realiza 6 casos de estudios centrados en empresas de Europa y dos ciudades finlandesas, incluidas empresas textiles, de cocina y fabricante de accesorios de baño, proveedor de servicios de TI y servicio de reciclaje; en Yen y Ng (2003) una industria textil en Hong Kong; en Suwisuthikasem y Tangsripairoj (2008) el Departamento de Ingresos de Tailandia y finalmente en Lagzian y Naderi (2014) una compañía de electricidad en Irán. Mientras que para la academia se han encontrado 15 artículos, en los cuales están Kreuzer et al. (2013); Pintarić et al. (2010); Macario y Srirama (2013); Dennehy y Sammon (2015); Cuylen et al. (2016); Sungkur, Gangabaksh y Rutah (2016); Taylor, Chau, Hui, Chau y Hui (2009); Drossos, Giaglis, Vlachos, Zamani y Lekakos (2013); Tung (2004); Diniz, Porto de Albuquerque y Cernev (2011); J. Chiu y Lai (2007); Matus et al. (2017); Radeski (2011); Tanna et al. (2005); Veselá y Radiměšský (2014); en países europeos. Adicionalmente en la industria se obtuvieron 12 trabajos con implementación de notificaciones, alertas, y comercio electrónico. Finalmente, se realizaron experimentos controlados en Lian (2015); Matus et al. (2017); Voutilainen y Pento (2003); Kim y Rohmer (2012); Kiatruangkrai et al. (2010); Yen y Ng (2003); Lagzian y Naderi (2014); Alippi et al. (2005); Taylor, Leonard y Cronan (2009); Tseng, Chang y Tseng (2007) y Novillo (2014).

Criterio de extracción	Estudios
Modelos de servicio	(Matus et al., 2017) (Sungkur et al., 2016) (Kwok, Laredo, & Maradugu, 2008)
Modelo de desarrollo	(Rofhök-björni, 2006) (Kaliontzoglou et al., 2006) (Spanic et al., 2011) (Vanjak et al., 2008) (Lian, 2015) (Ghazal et al., 2016) (Astudillo & Zambrano, 2006) (Sungkur et al., 2016) (Radeski, 2011) (Tseng et al., 2007) (Ji et al., 2014) (Carpio & Faicán, 2009)
Tipo de institución	(Pintarić et al., 2010) (Penttinen & Hyytiäinen, 2008) (Matus et al., 2017) (Suwisuthikasem & Tangsripairoj, 2008) (Chu et al., 2014) (C. H. Chiu et al., 2007) (Kiatruangkrai et al., 2010) (Rofhök-björni, 2006) (Spanic et al., 2011) (Vanjak et al., 2008) (Brun & Lanng, 2006) (Lian, 2015) (Ghazal et al., 2016) (Astudillo & Zambrano, 2006) (Alippi et al., 2005) (Kreuzer et al., 2013) (Cuylen et al., 2016) (Sungkur et al., 2016) (J. Chiu & Lai, 2007) (Taylor, Leonard, et al., 2009) (Novillo, 2014) (Kwok et al., 2008) (Carpio & Faicán, 2009) (Legner & Wende, 2006) (Matson & Ulieru, 2006) (Andreeff et al., 2001)
Partes que incluye	(Astudillo & Zambrano, 2006) (C. H. Chiu et al., 2007; Kiatruangkrai et al., 2010; Penttinen et al., 2009) (Alippi et al., 2005; Andreeff et al., 2001; Ghazal et al., 2016; Ji et al., 2014; Kaliontzoglou et al., 2006; Kim & Rohmer, 2012; Kreuzer et al., 2013; Lian, 2015; Macario & Srirama, 2013; Rofhök-björni, 2006; Sungkur et al., 2016; Tseng et al., 2007; Voutilainen & Pento, 2003) (Carpio & Faicán, 2009)
Usos	(Andreeff et al., 2001; Cuylen et al., 2016; Lee, Ju, & Jeong, 2006; Matus et al., 2017; Radeski, 2011; Suwisuthikasem & Tangsripairoj, 2008; Vanjak et al., 2008; Yen & Ng, 2003)

Tipos de arquitectura	(Astudillo & Zambrano, 2006; Brun & Lanng, 2006; Carpio & Faicán, 2009; C. H. Chiu et al., 2007; Chu et al., 2014; Ghazal et al., 2016; Ji et al., 2014; Kaliontzoglou et al., 2006; Kiatruangkrai et al., 2010; Kwok et al., 2008; Macario & Srirama, 2013; Matus et al., 2017; Radeski, 2011; Sungkur et al., 2016; Suwisuthikasem & Tangsripairoj, 2008; Tanna et al., 2005; Tseng et al., 2007; Vanjak et al., 2008)
Metodología de desarrollo utilizado	(Astudillo & Zambrano, 2006; Carpio & Faicán, 2009)
Lenguaje de programación (Back End)	(Matus et al., 2017; Suwisuthikasem & Tangsripairoj, 2008) (Astudillo & Zambrano, 2006; Brun & Lanng, 2006; Carpio & Faicán, 2009; Ji et al., 2014; Kiatruangkrai et al., 2010; Spanic et al., 2011; Sungkur et al., 2016; Tseng et al., 2007)
Lenguajes de programación (Front End)	(Astudillo & Zambrano, 2006; Carpio & Faicán, 2009; Ji et al., 2014; Kiatruangkrai et al., 2010; Spanic et al., 2011; Sungkur et al., 2016)
Base de datos	(Astudillo & Zambrano, 2006; Carpio & Faicán, 2009; Kaliontzoglou et al., 2006; Radeski, 2011; Spanic et al., 2011; Sungkur et al., 2016)
Atributos de seguridad	(Brun & Lanng, 2006; C. H. Chiu et al., 2007; Chu et al., 2014; Cuylen et al., 2016; Kaliontzoglou et al., 2006; Kiatruangkrai et al., 2010; Lee et al., 2006; Netter & Pernul, 2009; Novillo, 2014; Rofhök-björni, 2006; Spanic et al., 2011; Tanna et al., 2005; Taylor, Chau, et al., 2009; Vanjak et al., 2008)
Como se abordan los estudios	(Andreeff et al., 2001; Carpio & Faicán, 2009; C. H. Chiu et al., 2007; Ghazal et al., 2016; Kaliontzoglou et al., 2006; Kiatruangkrai et al., 2010; Lian, 2015; Rofhök-björni, 2006; Taylor, Leonard, et al., 2009; Tseng et al., 2007; Voutilainen & Pento, 2003)
Métodos de validación	(Alippi et al., 2005; Astudillo & Zambrano, 2006; Kiatruangkrai et al., 2010; Kim & Rohmer, 2012; Lagzian & Naderi, 2014; Lian, 2015; Matus et al., 2017; Taylor, Leonard, et al., 2009; Tseng et al., 2007; Voutilainen & Pento, 2003; Yen & Ng, 2003) (Novillo, 2014)
Tipo de estudio	(Alippi et al., 2005; Astudillo & Zambrano, 2006; Brun & Lanng, 2006; Carpio & Faicán, 2009; C. H. Chiu et al., 2007; Chu et al., 2014; Drossos et al., 2013; Ghazal et al., 2016; Ji et al., 2014; Kaliontzoglou et al., 2006; Kiatruangkrai et al., 2010; Kim & Rohmer, 2012; Lagzian & Naderi, 2014; Lian, 2015; Macario & Srirama, 2013; Matson & Ulieru, 2006; Matus et al., 2017; Penttinen et al., 2009; Penttinen & Hyytiäinen, 2008; Radeski, 2011; Rofhök-björni, 2006; Sungkur et al., 2016; Tanna et al., 2005; Taylor, Leonard, et al., 2009; Tseng et al., 2007; Vanjak et al., 2008; Veselá & Radiměšský, 2014; Voutilainen & Pento, 2003)

Tabla 3-F: Estudios por criterios de extracción.

El uso de computación en la nube para brindar servicios gubernamentales electrónicos proporciona varios beneficios, como una reducción en el costo total, almacenamiento de datos distribuidos, escalabilidad, responsabilidad, capacidad de modificación y administración de seguridad (Smitha, Chitharanjan, & Thomas, 2012). Sin embargo, estos servicios de gobierno electrónico basados en computación en la nube, asumen ciertos riesgos tangibles (acceso, disponibilidad, infraestructura e integridad) e intangibles

(confiabilidad, seguridad, confidencialidad y privacidad de datos, entre otros) (Zissis & Lekkas, 2011).

Los países con mayor cantidad de recursos están a la vanguardia de este tipo de desarrollo, ya que la adopción del gobierno electrónico por parte de un determinado grupo de personas está influenciado por varios factores, como la prosperidad económica, los niveles de educación, la estabilidad política y la aceptación cultural (Evans & Yen, 2006).

De esta manera, Khasawneh, Rabayah y Abu-shanab (2013a) menciona uno de los mayores desafíos para el gobierno electrónico, donde indica que las implementaciones de tecnología de la información a gran escala son proyectos arriesgados, ya que se multiplican los desafíos y riesgos al contar con la participación de varias organizaciones, como suele ser el caso en iniciativas de gobierno electrónico. También menciona que algunos de los riesgos pueden ser abordados alineándose cuidadosamente a las motivaciones de los socios. Por lo que, Khasawneh, Rabayah y Abu-shanab (2013b) propone un marco de alineación estratégica, el mismo que se usó para examinar las motivaciones de un grupo de agencias y compañías gubernamentales con el fin de minimizar dichos riesgos.

En cuanto a beneficios que las organizaciones pueden esperar y plasmar al cambiar a facturación electrónica tenemos: captura de facturas digitales (es decir, menos mano de obra), validación automatizada de facturas (es decir, menos errores), reducción de costos, administración de efectivo mejorada (es decir, trabajo rápido), optimización del sistema de TI, mejor relación con los proveedores, seguridad, aceptación de múltiples formatos (Ali, 2016).

Estudio demográfico

La Figura 3-7 muestra el número de artículos por país, donde Finlandia se puede ver como uno de los países donde el tema se ha abordado más ampliamente; sin embargo, Ecuador no se queda atrás, ya que ha aumentado la implementación y la investigación relacionada con la facturación electrónica, el comercio electrónico y las notificaciones o alertas al consumidor.

Amenazas de validez

Debido a la gran cantidad de bibliotecas digitales de renombre, solo se han considerado las cinco más importantes en el campo de la informática. Además, para obtener mejores resultados, se realizó una búsqueda manual en conferencias y revistas en el área de comercio electrónico. Sin embargo, la cadena de búsqueda se ha definido de acuerdo con los criterios de los investigadores, lo que puede requerir la inclusión de la bibliografía en términos relacionados con la facturación electrónica.

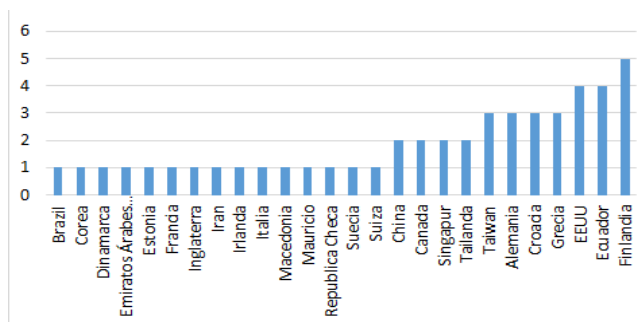


Figura 3-7: Distribución demográfica por países



Capítulo 4

Método para desarrollo de sistemas de información de pago

Este capítulo aborda la contribución principal de este trabajo de titulación, con el diseño de un método para el desarrollo de sistemas de información de pagos a organizaciones nacionales y envío de notificaciones a clientes, con información de pagos vencidos mediante medios digitales.

4.1 Contexto

El método propuesto está contextualizado como una solución que utiliza tecnologías de computación en la nube y proporciona agilidad a las organizaciones que hagan uso de los servicios provistos por la plataforma desarrollada con dicho método. En este sentido, las organizaciones se benefician al delegar a la plataforma las tareas de notificación sobre pagos vencidos de los clientes, generando un considerable decremento del tiempo utilizado por el personal de cobranzas para dichas labores. De igual manera, fomenta a los clientes a realizar sus pagos a tiempo, evitando caer en multas o suspensiones de servicio y por ende en las organizaciones ver reflejada una mayor agilidad en las operaciones de cobranzas para la recuperación de cartera.

Se han considerado para éste método, aspectos de seguridad dada su importancia en el entorno de computación en la nube, en el cual se proveen los servicios de notificación, generando confianza en las organizaciones que consuman dichos servicios.

4.2 Metodologías seleccionadas

Para el desarrollo de la metodología propuesta en este trabajo de titulación, se ha considerado necesario una alineación con la metodología propuesta por Chitforoush et al. (2007), y la inclusión de la metodología *ASD* de James A. Highsmith y Orr (2001). Estas metodologías fueron discutidas en la sección 2.4 respectivamente, en donde se tiene las comparativas respectivas con metodologías semejantes. De esta manera, las metodologías muestran buena compatibilidad para el desarrollo ágil con *MDA*, cubriendo el ciclo de vida de desarrollo por completo, junto con la inclusión de procesos de forma precisa y permitiendo definir el alcance del proyecto de forma adaptativa.

4.3 Estructura del método

En esta sección se muestra una descripción general de la estructura del método propuesto, a manera de dar una primera revisión rápida y precisa del mismo y poderlo comprender a mayor profundidad y detalle en secciones posteriores. De esta manera, se utilizó el lenguaje de meta-modelado *SPEM 2.0*, junto con sus especificaciones con el fin de poder representar de una manera gráfica las diferentes actividades que involucra el método propuesto.



4.3.1 Vista general

De esta manera en la Figura 4-1 se muestra el método propuesto, en donde se pueden observar las actividades principales junto con sus entradas y salidas respectivamente. Adicionalmente se puede ver qué actores son los que deben estar involucrados en cada una de las actividades del método. En adelante se utilizara *SPEM* para representar gráficamente las fases del método propuesto. La simbología utilizada se describe con detalle en el Anexo A.

4.4 Descripción del método

Una vez que se tiene claro cuáles son las actividades principales y una visión global del método se procede a profundizar a mayor detalle en cada una de las fases. En esta sección se muestra una descripción clara y concisa sobre el método junto a su diagrama en donde se observa claramente las actividades de cada fase, los roles de cada uno de los actores, las entradas y salidas que tiene cada actividad y cada fase como tal.

4.4.1 Fase de inicio del proyecto

En esta fase se identifican los objetivos específicos del proyecto, teniendo como base los objetivos generales del proyecto. La claridad de dichos objetivos es importante para estimar correctamente el alcance del proyecto. Para esto se realiza un análisis de requerimientos tempranos con el ingeniero de requerimientos junto con el experto de dominio y el director de proyecto. Se identifican los interesados, (en adelante denominados *stakeholders*) que están involucrados con el proyecto, son personas o instituciones internas o externas, que participan en las distintas fases del proyecto o cuyos intereses se podrían ver afectados de forma positiva o negativa durante la ejecución del proyecto, esto incluye clientes, patrocinadores, instituciones ejecutantes (públicas o privadas) y público en general (Jimenez, 2008). Es muy importante establecer los diferentes tipos de *stakeholders*, puesto que cada uno tiene su propio conjunto de requerimientos (Alexander & Stevens, 2002). Estos aspectos se muestran en la Figura 4-2.

Un entendimiento pobre del dominio es seguramente una causa de fracaso del proyecto. Para tener un entendimiento profundo del dominio se necesita entender los intereses, prioridades y habilidades de los actores involucrados en el dominio. Una representación del dominio resulta muy útil en etapas tempranas del proyecto donde los requerimientos pueden estar sujetos a ciertos cambios. Adicionalmente, mantener una representación estandarizada y bien organizada de conocimiento del dominio permite que éste sea compartido con otros dominios, facilitando el reúso. Conforme los sistemas en las organizaciones se interconectan e interoperan, es cada vez más importante entender cómo cooperan (entre sistemas y con humanos) para alcanzar las metas organizacionales y cumplir los intereses de los *stakeholders* (Yu, 1997).

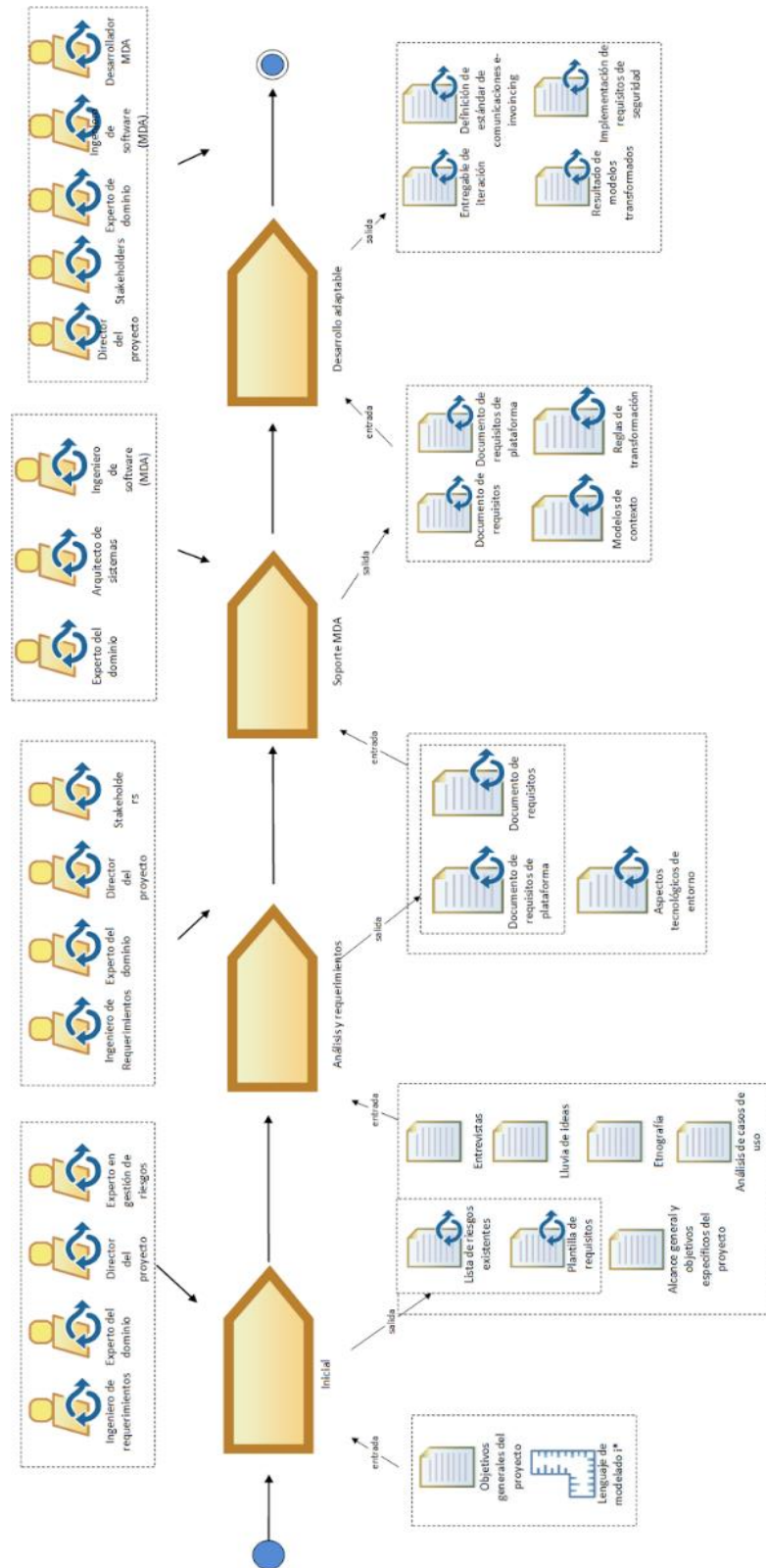


Figura 4-1: Vista general de método.

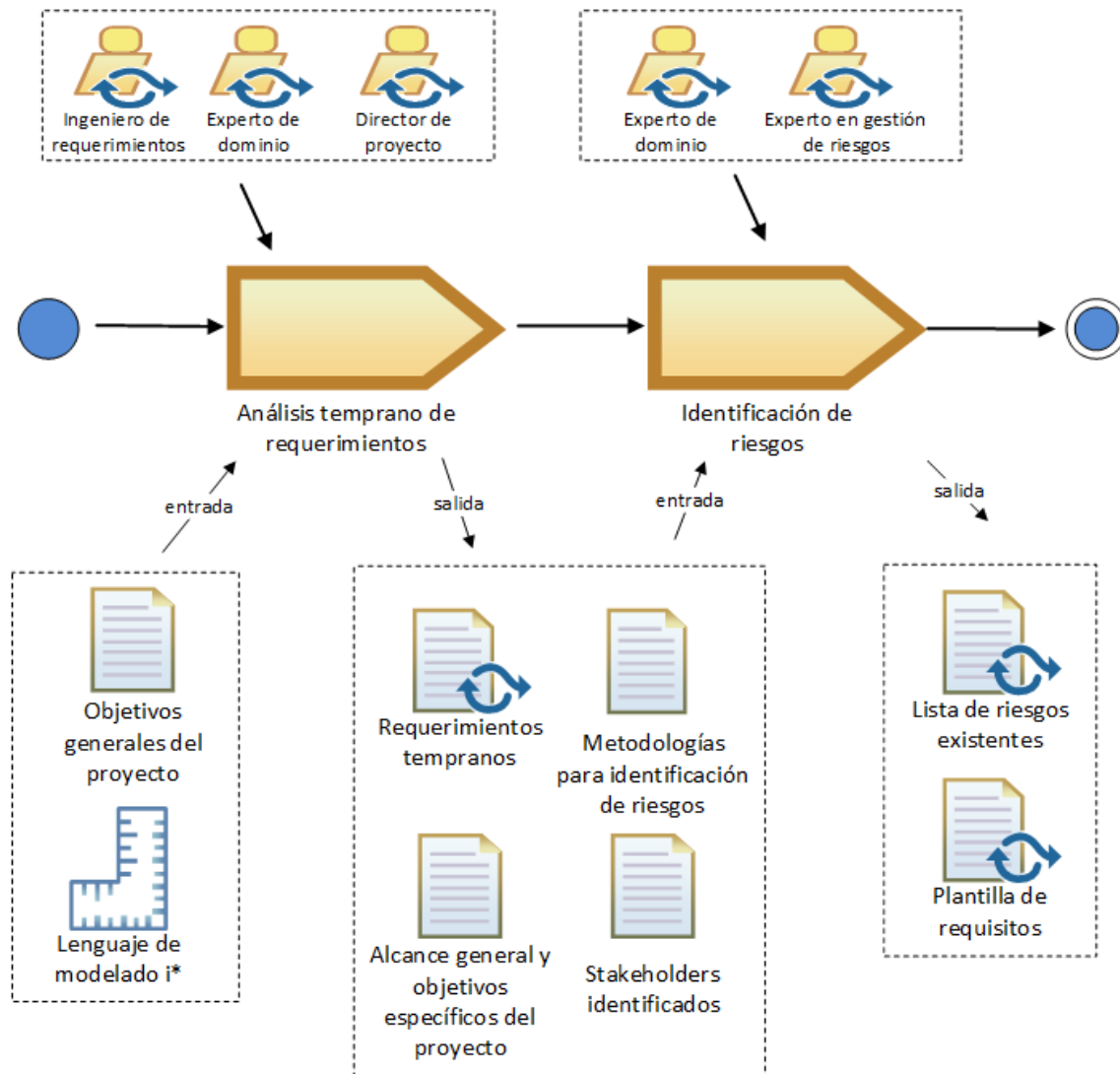


Figura 4-2: Fase de inicio de proyecto.

4.4.1.1 Análisis temprano de requerimientos

En esta fase se definen los requisitos tempranos. Según los objetivos generales del proyecto en esta actividad se utiliza el lenguaje de modelado i*, el cual facilita la exploración del dominio del proyecto, haciendo énfasis en aspectos sociales, mostrando de forma gráfica los actores del sistema, sus intenciones, dependencias y alternativas (Horkoff & Yu, 2016). El marco de trabajo con i* utiliza dos componentes de modelado principales (Yu, 1997):

- El modelo de dependencia estratégica (SD) es utilizado para describir las relaciones de dependencia entre los actores en un contexto organizacional o de proyecto.
- El modelo estratégico racional (SR) se utiliza para describir los intereses de los *stakeholders* y cómo pueden ser alcanzados por los sistemas.



4.4.1.2 Identificación de riesgos

Se identifican y se obtienen varios documentos y artefactos, como los requerimientos tempranos, alcance general y objetivos específicos del proyecto y los *stakeholders*. Por otro lado para dicha actividad, es necesario elegir la metodología adecuada o a gusto del profesional para la identificación de riesgos, con el propósito de aumentar la probabilidad y el impacto de eventos positivos, mientras se disminuye la probabilidad e impacto de eventos negativos para el proyecto. El experto en gestión de riesgos en caso de considerarlo necesario puede realizar un análisis más completo de riesgos, que en general contiene las siguientes actividades: (Jimenez, 2008)

- Identificar los riesgos: Se determinan los riesgos que presenta el proyecto, documentando sus características.
- Análisis cualitativo de riesgos: Consiste en asignar prioridades a los riesgos, evaluando la probabilidad de ocurrencia y el impacto asociado.
- Análisis cuantitativo de riesgos: se realiza un análisis numérico del efecto de los riesgos identificados.
- Planificación de respuesta a los riesgos: se definen acciones de respuesta para mejorar las oportunidades y disminuir las amenazas a los objetivos del proyecto.
- Monitorear y controlar los riesgos: se rastrean los riesgos encontrados, se monitorean los riesgos residuales, se identifican nuevos riesgos que pudieran surgir en el transcurso del proyecto y se evalúa la efectividad de aplicación de planes de respuesta frente a los riesgos. Esta tarea se la realiza durante todo el proyecto, específicamente en la fase de actividades umbrella.

Para este propósito se puede optar por utilizar una metodología reconocida formalmente. Entre las más destacadas se pueden mencionar: Modelo en espiral de Boehm et al. (1987); Método SEI-SRE de Williams, Pandelios y Behrens (1999); RISKIT de Kontio (1996); Método de PMI de Jimenez (2008).

4.4.2 Fase de análisis y requerimientos

Como se muestra en la Figura 4-3, en donde la fase realiza un análisis de requerimientos de la plataforma, incluyendo aspectos funcionales y no funcionales. Para esta tarea se involucran los miembros expertos de dominio, ingeniero de software *MDA* y los *stakeholders* o interesados identificados en la fase anterior.

Es importante que los requerimientos definidos en esta fase se tengan en cuenta como base para seleccionar un proceso de desarrollo de software que se aplicará en la fase de ejecución. Puede considerarse necesario realizar ciertas adaptaciones a dicho proceso, de manera que los requerimientos puedan ser implementados. En esta metodología se ha considerado la utilización de *ASD* como proceso de desarrollo de software y se han realizado ciertos ajustes para la integración en el entorno *MDA* que se plantea, lo cual se detalla en la fase de ejecución.

4.4.2.1 Elicitación de requerimientos

Para este propósito el ingeniero de requerimientos, junto con el experto de dominio y los *stakeholders*, realizan la actividad elicitación de requerimientos, teniendo en cuenta la

lista de riesgos existentes, una plantilla de requisitos, el alcance general y objetivos específicos del proyecto definidos en etapas previas. Como resultado se obtiene el alcance del proyecto definido en torno a los requisitos no funcionales, requisitos funcionales, requisitos de plataforma, requisitos de seguridad de la plataforma, requisitos de estándar de comunicación facturación electrónica y requisitos de despliegue de plataforma. En este contexto, se puede disponer de diferentes técnicas o herramientas como entrevistas, lluvia de ideas, etnografía y análisis de casos de uso.

4.4.2.2 Análisis y documentación de requerimientos

Para el Análisis y documentación de requerimientos es necesario disponer de los diferentes requisitos del sistema, en estos se encuentran los requisitos funcionales, no funcionales, de plataforma, de seguridad, de comunicación facturación electrónica, de pasarela de envío de SMS y servidor de correo. Finalmente se obtendrían todos los requisitos del sistema, así como de la plataforma. Una buena definición de requerimientos debe considerar todos los puntos de vista relevantes, dado que pueden presentarse conflictos de requerimientos que se contraponen por puntos de vista distintos de cada interesado o *stakeholder*. Por lo cual se recomienda resolver los requerimientos en conflicto antes de empezar con las siguientes fases, puesto que lidiar con este aspecto en medio del desarrollo resulta en un gran conflicto (Alexander & Stevens, 2002). Para esto se realiza un análisis de retroalimentación de los *stakeholders* sobre los requisitos definidos, y se realizan las correcciones necesarias en caso de existir, permitiendo obtener finalmente la documentación de requisitos del proyecto, incluyendo los requisitos de la plataforma.

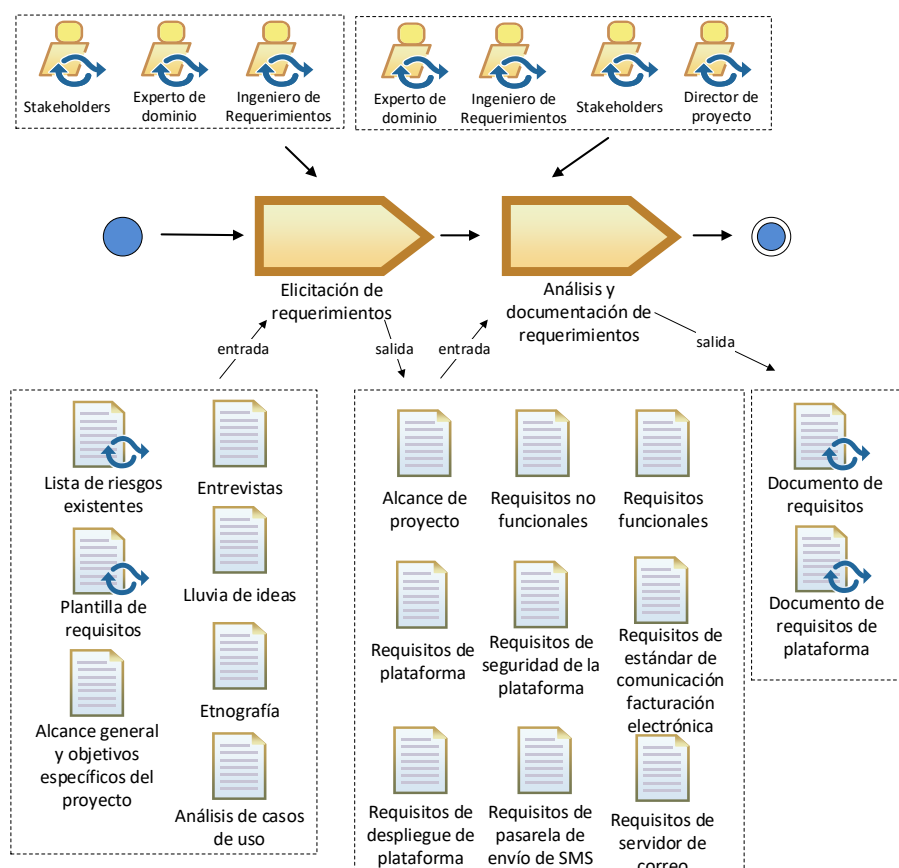


Figura 4-3: Fase de análisis y requerimientos.



A continuación se muestran algunos aspectos importantes a considerar en la ingeniería de requisitos con desarrollo de software utilizando métodos ágiles, conforme se propone en Paetsch, Eberlein y Maurer (2003):

Interacción de stakeholders

En el curso del proyecto es sumamente importante la interacción de los *stakeholders*, siendo un factor principal para la consecución de los objetivos del proyecto. Un punto clave de la metodología *ASD* consiste en la presencia de los *stakeholders*, accesibles oportunamente durante toda la ejecución del proyecto, a diferencia de las metodologías de desarrollo tradicionales, donde únicamente se interactúa con los *stakeholders* en etapas tempranas del proyecto.

Entrevistas

Es la técnica más común utilizada en la elicitación de requisitos, pues permite el acceso directo y sin restricciones al conocimiento requerido sobre el dominio evitando malos entendidos. Es importante que la entrevista sea directamente entre las partes involucradas, es decir ingeniero de requerimientos, experto de dominio y *stakeholders*, puesto que cuando al utilizar cadenas de transmisión de conocimiento cierta información puede verse perdida, alterada o inconsistente, afectando también la confianza de los *stakeholders* en el equipo de elicitación de requerimientos.

Priorización

Este término se encuentra en todos los enfoques de desarrollo ágil. La práctica común sugiere la implementación en primer lugar de las características con mayor prioridad, en busca de obtener primero los entregables con mayor valor para los *stakeholders*. Durante el desarrollo la comprensión y claridad del dominio y sus requisitos va creciendo y es posible agregar nuevos requerimientos sobre la marcha. Para mantener el orden de prioridades actualizado a la fecha es importante seguir el concepto de priorización constantemente.

Sesiones de diseño de aplicación conjunta

En *ASD* se utilizan este tipo de sesiones (*JAD* – Joint Application Design) para incrementar la participación de los *stakeholders*. Se busca tener una constante retroalimentación de los *stakeholders*, por lo cual estas sesiones se realizan frecuentemente durante el proceso de desarrollo. Cada una de éstas debe estar documentada y disponible para futuras referencias o consultas.

Modelado

En el desarrollo ágil normalmente se utilizan modelos para comunicar el entendimiento de partes específicas del sistema a desarrollarse, los cuales luego de haber cumplido su propósito son desechados. Por otro lado, en éste método se debe considerar la ingeniería de requisitos orientada a desarrollo ágil de software, de esta manera se pretende utilizar modelos con diferentes niveles de abstracción. Estos modelos deben formar parte de la documentación del sistema y estar siempre actualizados.

Documentación

En el desarrollo de software ágil normalmente la documentación no se realiza con rigor, puesto que crear documentos de requerimientos completos y consistentes es considerado



ineficiente. Esto justifica que los documentos en desarrollo ágil sean mucho más cortos que los enfoques de desarrollo tradicionales, sin embargo en el enfoque utilizado en éste método se hace una excepción.

La carencia de documentación puede causar problemas a largo plazo a los equipos de desarrollo. La documentación es utilizada para transmitir el conocimiento sobre los requerimientos, y mantenerlos disponibles ante cualquier consulta. Es una característica importante para la mantenibilidad, considerando que los miembros del equipo de desarrollo pueden ser cambiados con el paso del tiempo, siendo importante que los nuevos miembros puedan disponer de documentación clara sobre el proyecto o sistema a desarrollar.

Por otro lado, la documentación debe ser concisa y contemplar únicamente los aspectos más relevantes, facilitando la actualización frecuente en caso de presentarse cambios en el sistema. Se sigue la premisa que en métodos de desarrollo ágiles se debe emplear menos tiempo en la documentación de requerimientos en comparación a los métodos tradicionales de software.

Las ventajas o desventajas de un manejo de documentación están ligadas al tamaño del equipo de desarrollo. En equipos grandes se considera más conveniente contar con documentación en lugar de explicar los requerimientos a todo el personal.

Validación

Se realizan frecuentes reuniones de revisión de los requerimientos, verificando que el proyecto se encuentre encaminado correctamente. De esta manera, las preguntas que puedan surgir de los *stakeholders* pueden ser solventadas inmediatamente por el grupo de desarrollo, se pueden solicitar cambios si no se satisfacen los intereses o necesidades de dichos *stakeholders*.

Administración

Es importante que cuando se realicen cambios en la documentación de requerimientos se describa los motivos de dichos cambios, permitiendo la trazabilidad. Dado que esto requiere de mayor tiempo, la ingeniería de requerimientos para métodos ágiles propone utilizar un nivel menor de detalle en las descripciones de requerimientos, pudiendo incluso postergarse la claridad de ciertos aspectos de requerimientos para cuando sean necesarios en una iteración dada.

Observación y análisis social

A pesar de que esta técnica no suele mencionarse en enfoques de desarrollo ágil, es sumamente útil. Existen situaciones en las que la rutina del trabajo de los usuarios o *stakeholders* hace que no mencionen todos los detalles de sus procesos de trabajo, por lo cual el grupo de elicitación de requerimientos debe observar los detalles importantes que no hayan sido mencionados.

Requerimientos no funcionales

A pesar que en el enfoque ágil de desarrollo se cuenta con la interacción oportuna de los *stakeholders*, normalmente no es suficiente para recabar los requerimientos no funcionales. Este tipo de requerimientos suelen ser desconocidos o muy técnicos para los *stakeholders*, por lo cual, deben ser considerados aspectos de mantenibilidad, portabilidad,

seguridad o rendimiento, puesto que de estos factores depende la decisión de la elección de la base de datos, lenguaje de programación o sistema operativo.

4.4.3 Fase de soporte MDA

En esta fase es donde se toman decisiones sobre el desarrollo en cuanto a lo que *MDA* define y su arquitectura de alto nivel. Con el fin de identificar aspectos relevantes con la plataforma objetivo, lenguaje de modelado y transformaciones, además se modelan los diferentes modelos de contexto requeridos junto con sus reglas de transformación, acorde a la Figura 4-4 en donde se muestran las diferentes actividades a realizarse.

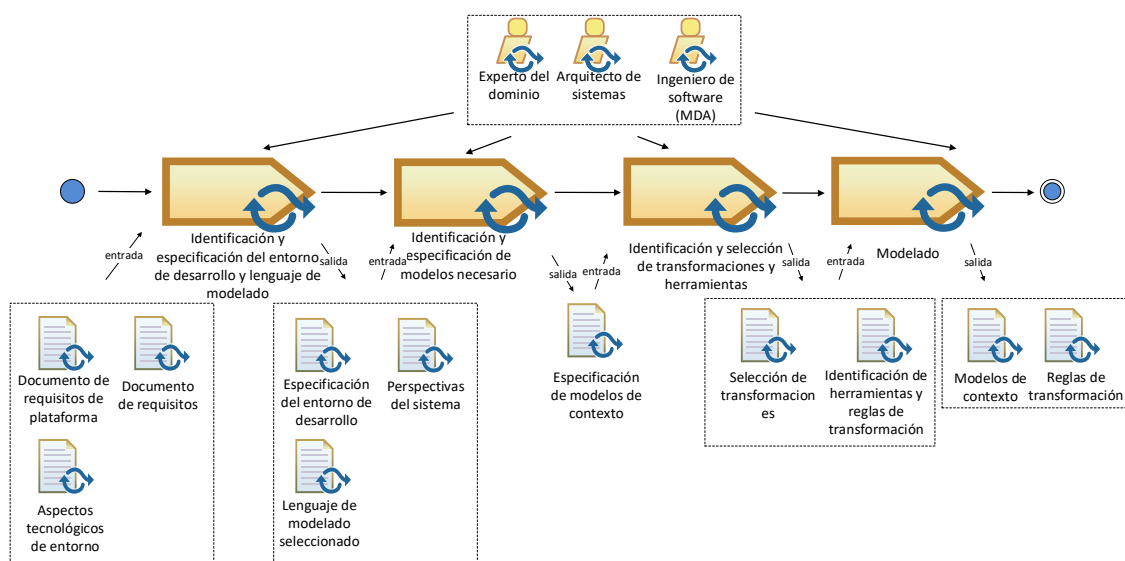


Figura 4-4: Fase de soporte MDA.

En esta fase se debe contar con expertos del dominio, arquitectos de sistemas e ingeniero de software (*MDA*). Las actividades a realizarse son las siguientes:

4.4.3.1 Identificación y especificación del entorno de desarrollo y lenguaje de modelado

La idea del desarrollo de *MDA* es la separación en diferentes puntos de vista a las especificaciones del sistema, según la implementación del mismo, de acuerdo a la plataforma a la cual se desee realizar el desarrollo. El objetivo de esta actividad es obtener e identificar de manera detallada las características específicas en cuanto a la plataforma en donde se va a implementar el sistema, de esta manera se debe incluir datos específicos en cuanto a la parte del hardware como del software, además se debe tener en cuenta los aspectos tecnológicos del entorno. De esta manera se tiene claro el panorama para poder realizar los modelos específicos de plataforma.

Es así que los modelos producidos deben ser precisos y lo suficientemente detallados para obtener una precisa generación de código. El lenguaje de modelado equivale a un lenguaje de programación de alto nivel. Para poder definir un lenguaje de modelado para el sistema a desarrollar primero se debe identificar las diferentes perspectivas de sistema a modelar (Quintero & Anaya, 2007). Para ello es necesario que los expertos del dominio se



involucran, así como ingenieros de software (*MDA*), con el fin de plasmar las diferentes funcionalidades, comportamientos y estructura del sistema en modelos como caso de uso, diagramas de estados y diagramas de clases, los mismo que son producto de la fase anterior. De acuerdo con estos modelos, se selecciona un lenguaje de modelado que sea lo suficientemente expresivo para el dominio y la situación del proyecto. Este lenguaje se usará en las transformaciones que se realicen respectivamente de modelos *PIM* y *PSM* (Kleppe, Warmer, & Bast, 2003).

4.4.3.2 Identificación y especificación de modelos necesarios

En esta fase se obtienen las especificaciones de los diferentes modelos de contexto necesarios para el desarrollo del sistema, teniendo como entrada a las especificaciones del entorno de desarrollo, el lenguaje de modelado y las diferentes perspectivas del sistema. Esto implica trabajar con expertos de dominio, arquitecto del sistema e ingeniero de software (*MDA*) para determinar que funcionalidad se incluirá en el sistema y cuál la ofrece el entorno del sistema. Finalmente se especificara que modelos de contexto son necesarios para realizar las transformaciones.

4.4.3.3 Identificación y especificación de transformaciones y herramientas

Existe una estrecha relación entre los diferentes modelos construidos en *MDA*. Mientras que los modelos más abstractos son la base para la elaboración de los modelos específicos, los modelos son los que soportan un nivel de abstracción mayor. En las transformaciones entre modelos se representa esta relación, las cuales constituyen la principal característica de *MDA*. A dichos procesos de transformación se les denomina “*mappings*”, los mismos que se encuentran conformados por varias reglas de transformación que son operativas en un dominio específico (Muñoz, Solarte, & Heredia, 2015).

Aquí se definen las diferentes transformaciones necesarias que se realizaran para el desarrollo del sistema. Se tiene de modelo a modelo, en donde se transforma de un modelo base a un modelo destino con menor abstracción, por otro lado se tiene de modelo a texto, en donde se transforma el modelo base a código fuente en el lenguaje necesario. Para ello se debe tener en cuenta el lenguaje de modelado seleccionado en el paso anterior. Después de decidir qué transformaciones se necesitan, se deben especificar en detalle, con las reglas de transformación y las anotaciones adecuadamente descritas con lo cual se realizará las transformaciones respectivas (Chitforoush et al., 2007)(Kleppe et al., 2003).

Es necesario el uso de la definición de herramientas de transformación cuando no se puede usar una definición de transformaciones y necesita crear sus propias definiciones de transformación. Para esto se requiere del uso de lenguajes de scripting específicos como Query/View/Transformation (*QVT*) (Kleppe et al., 2003).

Estas herramientas juegan un papel muy importante en el desarrollo basado en *MDA*, ya que ayudan en la creación o modificación de reglas de transformación de los distintos modelos. Para ello hay que tener en cuenta la compatibilidad de la herramienta y las funcionalidades que proporciona al entorno *MDA*. En el desarrollo impulsado por modelos,

los lenguajes de modelado, los modelos de plataforma y las especificaciones de transformación son todos elementos de reutilización.

4.4.3.4 Modelado

Una vez identificadas las diferentes características que deben tener los modelos a realizar, sus respectivas transformaciones, reglas de transformaciones y herramientas, se realiza respectivamente la implementación de los modelos de contexto necesarios para el sistema junto con sus reglas de transformación.

4.4.4 Fase de desarrollo adaptable

Finalmente en esta fase se puede observar la cooperación de la metodología ágil ASD descrita en la sección 2.4.3, para el desarrollo del método propuesto, como se muestra en la Figura 4-5 las tres fases de ASD especulación, colaboración y aprendizaje, reflejan la naturaleza dinámica que sigue este proceso, reemplazando el determinismo por el descubrimiento, siguiendo un comportamiento de adaptación continua a cambios, aprendizaje constante, re-evaluación e intensa colaboración entre desarrolladores, directores de proyecto y clientes.

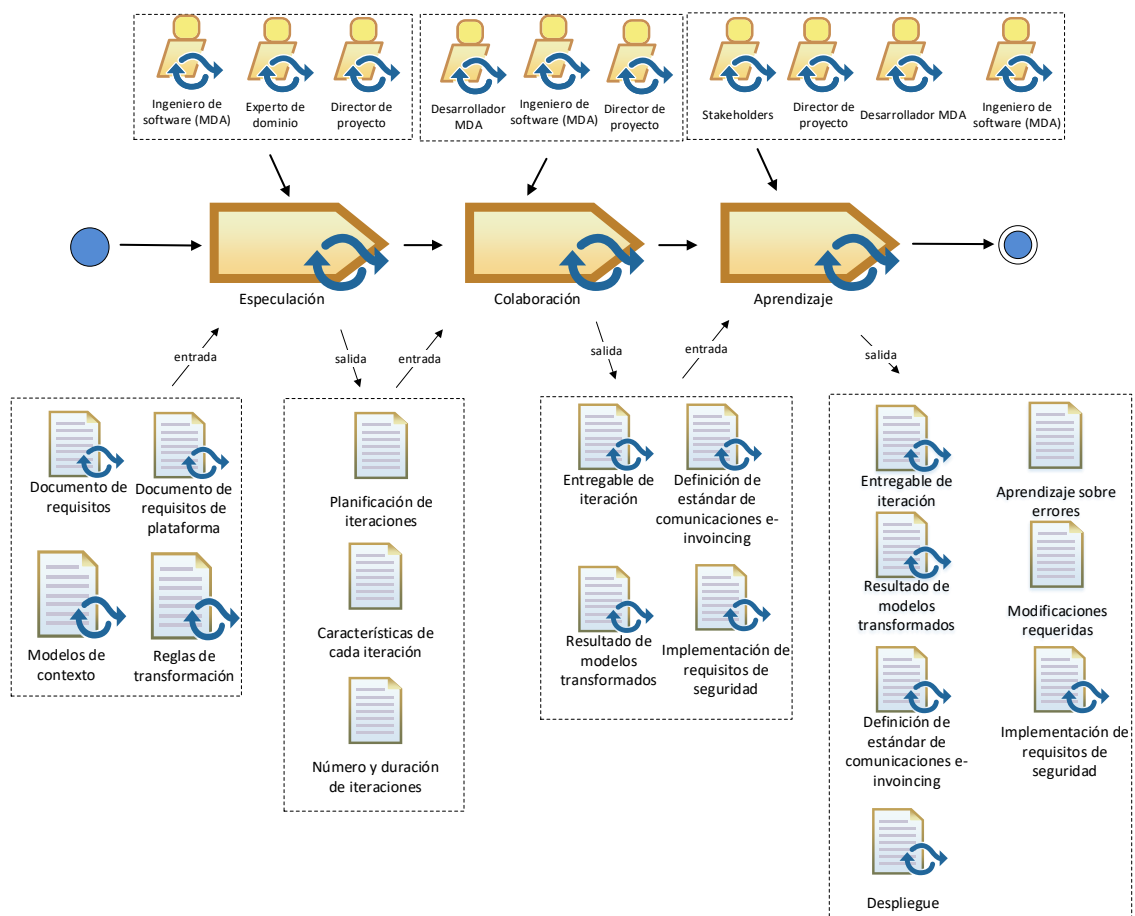


Figura 4-5: Fase de desarrollo adaptable.

Un ciclo de vida de ASD no está enfocado en tareas, sino en resultados que son identificados como características del sistema, y tiene seis características básicas:



- Enfocado a la misión

A pesar de que los requerimientos de un proyecto pudieran no estar del todo claros al inicio, es importante que la misión principal siempre esté bien definida. Por lo cual, la misión actuará como líneas de guía que incentiven la exploración pero a la vez provean la directriz del proyecto.

- Basado en características

Los ciclos de vida de *ASD* no se basan en tareas, sino en características, es decir, las funcionalidades que son desarrolladas durante una iteración. Dichas características evolucionan conforme se dan las iteraciones y la retroalimentación correspondiente por los interesados.

- Iterativo

Los ciclos de vida tienen naturaleza iterativa, y deben venir acompañados de retroalimentación en cada fin de ciclo, para permitir el aprendizaje y corregir el rumbo en caso de ser necesario.

- Tiempo de entrega definido

Cada iteración en *ASD* debe tener definido un tiempo de entrega; sin embargo, no se refiere a la utilización de fechas límite que puedan provocar competencia en el equipo de trabajo o vulnerar la calidad de los entregables. Por el contrario, se pretende marcar el paso, especialmente en entornos muy inciertos donde la tasa de cambios es muy alta.

- Riesgos gestionados

Es necesario que se identifiquen y se evalúen los riesgos críticos en cada iteración de *ASD*.

- Tolerante a cambios

El enfoque de *ASD* es orientado a cambios por lo cual el cambio debe ser visto como la habilidad de incorporar ventaja competitiva, más no un problema en el desarrollo.

El proceso de *ASD* se desarrolla en tres fases, dispuestas como se muestra en el esquema de la Figura 4-6 y se describe a continuación (tutorialspoint.com, 2016):

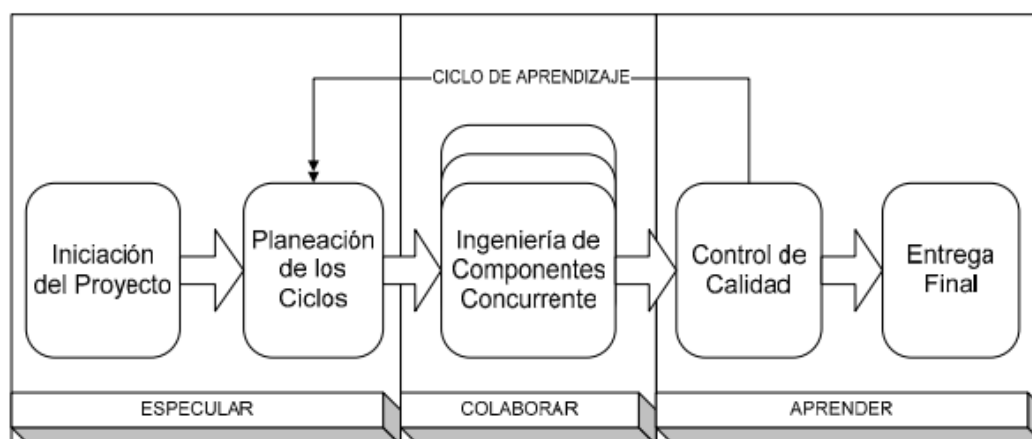


Figura 4-6: Proceso ASD (James A. Highsmith & Orr, 2001).

4.4.4.1 Especulación

El término planeación resulta demasiado determinista pues pretende indicar con alto grado de certeza el resultado deseado. El dirigir un proyecto conforme un plan restringe la habilidad del director de proyecto para guiar el proyecto en direcciones innovativas posiblemente mejores. Es por esto que en *ASD* se reemplaza el término planeación y se utiliza especulación en su lugar. En este caso no se abandona la planeación, pero se hace énfasis en reconocer ciertos problemas complejos que dan paso a la exploración y experimentación. Este hecho es factible al dividir la totalidad del proyecto utilizando ciclos cortos de desarrollo (tutorialspoint.com, 2016).

Basándose en el alcance, conjunto de requisitos de proyecto y de plataforma, modelos de contexto, reglas de transformación, disponibilidad de recursos, se debe establecer el número de iteraciones necesarias y la duración de cada una de ellas. Esto debe realizarse en acuerdo común del ingeniero de software *MDA*, el experto de dominio y el director de proyecto. De esta manera, en proyectos pequeños se suelen utilizar iteraciones de dos semanas, y en proyectos de mayor tamaño se utilizan iteraciones de cuatro a ocho semanas. Una vez se han definido estos aspectos se debe calendarizar cada iteración. Para cada iteración se asigna un objetivo, que consista en entregar un conjunto de características o entregables, que sean visibles para demostrar su funcionalidad a los revisores y poder recibir retroalimentación.

Los desarrolladores junto con los clientes asignan las características a cada iteración, teniendo en cuenta estimaciones de tiempo, riesgos o dependencias. Los *stakeholders* pueden proponer la priorización de ciertas características.

4.4.4.2 Colaboración

El construir sistemas complejos requiere que un gran volumen de información sea recolectado, analizado y aplicado en la resolución de los desafíos que se presenten. Por ende, dicha información puede verse distribuida en los diferentes miembros del equipo de trabajo, permitiendo que diversos conocimientos trabajen en conjunto por medio de la colaboración, compartiendo el conocimiento para tomar decisiones acertadas o solucionar problemas técnicos.



- Colaboración para equipos distribuidos

Se debe considerar diversidad en cada equipo, la forma en la que las personas interactúan y la forma en la que se gestionan las interdependencias.

- Colaboración para proyectos más pequeños

En proyectos donde existe proximidad física entre los miembros del equipo, resultan efectivas las charlas informales de pasillo o garabatos de pizarra.

- Colaboración para proyectos grandes

Cuando los proyectos tienen gran tamaño se suelen utilizar herramientas de colaboración y constante interacción con el director de proyectos.

En esta sección el desarrollador *MDA*, previa coordinación con el ingeniero de software *MDA* y el director de proyecto realiza la transformación de los modelos de contexto establecidos en etapas anteriores. Adicionalmente se realiza la definición del estándar de comunicaciones de facturación electrónica que permitirá el intercambio de información de pagos entre las empresas y la plataforma propuesta en este trabajo de titulación. Es importante que dicho estándar tenga compatibilidad tecnológica, permitiendo que cualquier empresa pueda comunicar sus sistemas con la plataforma dispuesta en la nube. Adicionalmente en esta fase se dota a la plataforma de los requisitos de seguridad establecidos en la fase de análisis y requerimientos.

4.4.4.3 Aprendizaje

Este segmento consiste en incrementar el conocimiento del equipo de trabajo, utilizando revisiones técnicas, retrospectivas de proyecto o grupos de enfoque con los clientes. Es importante que las iteraciones sean cortas, de esta manera se obtendrá aprendizaje de errores pequeños antes que errores grandes. El aprendizaje puede ser sobre el producto desarrollado o sobre la forma en la que dicho producto de software está siendo desarrollado.

- Revisión de calidad desde la perspectiva del cliente

Recibir retroalimentación del cliente en ASD es la principal prioridad, para lo cual se utilizan reuniones de enfoque, donde explora el modelo de trabajo y se registran las peticiones de cambios por parte del cliente.

- Revisión de calidad desde una perspectiva técnica

Resulta importante realizar revisiones periódicas sobre aspectos técnicos, esto incluye revisiones del código fuente y arquitectura del sistema. Las primeras se suelen realizar de forma continua, las segundas se pueden realizar semanalmente o al final de cada iteración.

- Rendimiento del equipo de desarrollo y de las prácticas utilizadas

El equipo debe monitorear su propio rendimiento de forma periódica. Esta retrospectiva permite aprender de sí mismos y de su trabajo como equipo, de esta manera se determina lo que no está funcionando, lo que se requiere reforzar y en lo que se requiere disminuir esfuerzos.

- Estado del proyecto



Una revisión del estado o progreso del proyecto permite la planeación de ciclos futuros de trabajo. Para esto se analizan las características ya desarrolladas, y se responde la pregunta ¿Qué se ha aprendido en esta iteración? ¿Qué ha cambiado de nuestra perspectiva hacia dónde vamos?

Finalmente, se realiza el despliegue en la nube de todos los productos obtenidos en la etapa de colaboración, una vez que hayan sido revisados y aprobados por los stakeholders, director de proyecto, desarrollador MDA e ingeniero de software MDA.



Capítulo 5

Implementación

Este proceso lleva a la práctica el método presentado. Aquí se ilustra mediante una instanciación llevada a la vida práctica cada una de las fases que intervienen en el método propuesto.

5.1 Fase Inicial

Según la sección 4.3.1; en la primera fase se realiza un análisis de requerimientos tempranos, los objetivos principales y secundarios, el alcance, identificación de los actores implicados (*stakeholders*), riesgos, entre otros. Así, como el contexto del sistema y modelado i*. De esta manera se tiene un primer acercamiento a los diferentes requisitos del sistema y hacia donde tiene que ir dicho desarrollo.

5.1.1 Requerimientos tempranos

Para la elicitación de requerimientos tempranos se ha utilizado el marco de trabajo I-Estrella y en particular se ha seguido el modelo de agente-meta propuesto por Paetsch et al. (2003) para establecer las actividades de propósito y elicitación, creación del modelo y análisis. A pesar que el modelo sugiere pasos ordenados, puede ser necesaria cierta flexibilidad de manera que se permita iteración, especialmente en casos donde la interacción con los *stakeholders* se presenta al transcurrir tiempos considerables, recibiendo una retroalimentación que dé paso a cambios que deberán ser correctamente actualizados en el modelo. A continuación, se describen las actividades del modelo, junto con los resultados de su implementación.

5.1.2 Propósito y elicitación

En esta actividad se identifica el alcance y propósito del proceso de modelado. El alcance de este trabajo de titulación contempla el desarrollo de un sistema de información de pagos a organizaciones ecuatorianas, que permita gestionar comprobantes electrónicos y enviar información de pago a sus clientes a través de medios digitales. Por ende, el propósito del proceso de modelado es mostrar la perspectiva de agentes involucrados en la utilización directa o indirecta de la plataforma a desarrollar, junto con las metas que confieren a cada uno de dichos elementos, incluyendo participantes que dan soporte al proceso de funcionamiento de la plataforma.

5.1.3 Creación del modelo

Para la creación del modelo es necesario considerar los siguientes aspectos:

5.1.3.1 Contexto del sistema

Faceta de sujeto

El sistema a desarrollar debe estar basado en tecnologías de computación en la nube y permitir escalabilidad, de manera que las organizaciones que requieran de los servicios provistos puedan incrementar en cantidad sobre la marcha. Por otro lado, se requiere la utilización de una infraestructura de computación en la nube que permita mantener seguros



los datos de las organizaciones, disponiendo de medidas de prevención y contingencias ante cualquier emergencia o desastre.

Los servicios web expuestos por la plataforma en la nube, procesarán las peticiones que se realicen desde sistemas informáticos *ERP* de cada una de las organizaciones o empresas. Para este propósito se requiere definir un estándar que será el utilizado para estructurar la información de dichas peticiones. Este formato deberá ser utilizado para solicitar las siguientes operaciones en los servicios web:

- Registrar o actualizar información de la empresa

Para poder utilizar los servicios de la plataforma, las empresas deberán registrar su información de contacto, que consiste en: ruc, razón social, dirección, nombre comercial, teléfono, correo. Adicionalmente se deberá registrar la plantilla de notificación de correo, es decir: título, descripción, descripción adicional y frecuencia de notificación (opciones pre-establecidas). De igual manera, para la plantilla de notificación de mensajes de texto se deberá registrar título, descripción, frecuencia de notificación (opciones pre-establecidas). Esta operación podrá ser ejecutada en reiteradas ocasiones, cumpliendo la función de actualizar la información, para lo cual se deberá utilizar el formato conforme el estándar definido.

- Recepción de comprobantes para notificación

La plataforma recibirá los comprobantes (facturas electrónicas) junto con la información de contacto necesaria para el despacho de notificaciones a los clientes, es decir número de teléfono celular y dirección de correo electrónico. Adicionalmente se incluirá la información sobre el número de pagos de dicho comprobante y el valor asociado a cada pago, junto con las fechas de caducidad y plazo máximo de cada pago. Para este propósito se utilizará el estándar definido, pudiendo solicitar la recepción de múltiples comprobantes en una misma solicitud.

- Confirmación de pagos

Las empresas podrán realizar la confirmación de los pagos que ya hayan sido cancelados o que deseen dar de baja, para lo cual realizarán una petición a los servicios web de la plataforma utilizando el formato del estándar definido, en el que se incluya la clave de acceso del comprobante en cuestión, y los identificadores de los pagos que se desean confirmar sobre dicho comprobante. El formato permite incluir la confirmación de pagos de múltiples comprobantes en una misma solicitud.

Para cada petición recibida por el servicio web de la plataforma se verificará la procedencia de la petición, para lo cual se utilizarán certificados digitales permitiendo la autenticación bidireccional (empresa-plataforma y plataforma-cliente). Los certificados utilizados deberán ser emitidos por la autoridad de certificación correspondiente en Ecuador, es decir, el Banco Central del Ecuador. Adicionalmente, se deberá consumir las operaciones del servicio web utilizando un canal seguro, que permita la transmisión cifrada de las comunicaciones.



Las respuestas correspondientes a cada petición o invocación de una operación de los servicios web de la plataforma deberán contener información referente a dicha operación, es decir, una lista de avisos que contengan la clave de acceso (identificador único de cada factura), el identificador del pago al que se hace referencia y la descripción o información del resultado de la operación.

La plataforma ejecutará el despacho de notificaciones en las frecuencias de ocho horas, doce horas, diariamente, dos veces a la semana y una vez a la semana. Estos procesos se ejecutarán de forma automática, sin afectar la disponibilidad de los servicios expuestos por la plataforma, para lo cual se utilizará paralelismo.

Faceta de uso

El cliente será cualquier organización o empresa nacional que solicite, y haga uso del servicio el cual realiza la gestión de sus facturas electrónicas así como el envío de información hacia sus respectivos clientes por medio de notificaciones por medio de medios digitales, tales como correo electrónico y mensajes de texto. El desarrollo de la plataforma no contempla el desarrollo ni modificaciones de sistemas de terceros como sistemas informáticos *ERPs* por parte de las organizaciones o empresas.

La plataforma brindará el servicio mencionado por medio de servicios web, que responden a peticiones *SOAP*, para el registro de los datos de cada organización, así como sus preferencias en cuanto a plantillas para las notificaciones; un segundo servicio para la subida de facturas hacia la plataforma y finalmente el dar de baja a pagos que ya se hayan cancelado por parte de los respectivos clientes.

Faceta de plataforma

Se utilizará una arquitectura cliente - servidor, en donde las organizaciones consumen los diferentes servicios ofrecidos por parte de la plataforma.

- **Servidor**

Se requiere de un servidor, el cual receptorá todas las peticiones por parte de las organizaciones que contraten el servicio para realizar todos los procesos de gestión y notificación de los servicios web ofrecidos a las diferentes organizaciones.

- **Especificaciones técnicas del servidor:**

Sistema Operativo

Linux Ubuntu Server 16.04

Procesador

AMD de 64 bits con doble núcleo

Memoria

4GB memoria RDIMM DDR3

Discos Duros

Elastic Block Store



Faceta de desarrollo

Respecto al desarrollo de la solución, se utilizará el lenguaje de programación Java para implementar los servicios web a ser desplegados en la plataforma. Estos servicios deberán responder a peticiones de tipo *SOAP*. El uso del protocolo *SOAP* permite enfocar el desarrollo en el procesamiento del lado del servidor, dado que no se requiere la transferencia de recursos sino el procesamiento de los datos de los mensajes enviados a los servicios web (www.chakray.com, 2016). Las operaciones cuentan con interfaces que solicitan la información correspondiente a dicha petición, como se describe en la faceta de sujeto, para lo cual se deberá definir un estándar en lenguaje *XML*. Las respuestas que generen los servicios web consistirán en objetos de tipo (*POJO - Plain Old Java Object*) que contienen la información referente a la ejecución de la operación correspondiente, conforme se describe en la faceta de sujeto.

Para la autenticación entre las empresas y el servidor, y viceversa, se utilizarán certificados digitales X.509 emitidos por la autoridad competente, en este caso, el Banco Central del Ecuador. Estos certificados serán utilizados para mantener una conexión segura *HTTP* mediante *SSL*, brindando confidencialidad y autenticidad entre las partes de la comunicación. En cada petición recibida por la plataforma se realizará una verificación de la procedencia del certificado asociado a la conexión de dicha petición, para lo cual se usará la herramienta *OpenSSL* que verificará que dicho certificado proceda y esté firmado por la autoridad de certificación Banco Central del Ecuador.

La base de datos que utilizará la plataforma para el almacenamiento de toda la información generada y procesada en los servicios web será *MySQL 5.7*. La capa de persistencia del servidor hará uso de la herramienta de mapeo objeto-relacional *Hibernate* para realizar las operaciones de creación, lectura, actualización y eliminación en la base de datos. Se utilizará un pool de conexiones *CP80* para la gestión de concurrencia en acceso a la base de datos.

Con respecto a las comunicaciones con la pasarela de envío de mensajes de texto, se utilizará el protocolo *HTTP* para realizar solicitudes de tipo *POST*, las cuales incluirán las credenciales de acceso proporcionadas por la pasarela. Por otro lado, para el despacho de notificaciones de correo electrónico se utilizará el protocolo *SMTP* con autenticación mediante credenciales establecidas previamente en la cuenta de correo electrónico.

5.1.3.2 Diagramas de modelado I-Estrella

En esta sección se tendrá que identificar correctamente a los diferentes actores que están involucrados, ya sea de forma interna como externa a la plataforma. Así, como la realización de los diferentes modelos de dependencias estratégicas (SD), junto con su descripción. En adelante se utilizarán los diagramas de modelado *i**, para definir el contexto de sistema. Una descripción detallada de este tipo de diagramas se muestra en el Anexo C.

Actores del sistema

En la Tabla 5-A se muestran a los diferentes actores que estarán involucrados, tanto interno como externos, de tipo persona, hardware y software.

Actor		Descripción
Persona/Entidad		
Organización		Corresponde a todas las organizaciones, empresas o compañías que utilicen los servicios de la plataforma, por medio de sus sistemas ERP.
Cliente		Consiste en todos los clientes de las organizaciones, los cuales reciben las notificaciones de SMS o de correo por los saldos o deudas que mantienen con las organizaciones.
Hardware		
Plataforma		Es la parte principal la arquitectura, pues consiste en el servidor que provee las operaciones de registro de organizaciones, envío de comprobantes y confirmación de pagos. Esto incluye la base de datos.
Software		
Pasarela de SMS		Corresponde al proveedor de servicio de envío de mensajes de texto. En este trabajo, permite el envío de notificaciones de tipo SMS hacia los clientes.
Servidor de correo		Es el servidor que provee los servicios correspondientes a correo electrónico. En este caso, para el despacho de notificaciones de correo electrónico hacia los clientes.

Tabla 5-A: Actores del sistema.

Descripción del modelo SD

A continuación, se muestra las descripciones de las diferentes dependencias entre los actores del sistema definidos previamente, junto con su modelo respectivo.

Dependencia entre plataforma y organización

En la Figura 5-1 se puede observar la relación en donde la plataforma depende de la organización para que sean enviados los datos, plantillas de notificación e información de los pagos a notificar. También la plataforma depende de la organización para que los pagos sean confirmados. Para esto se requiere que los datos sean llenados conforme los estándares establecidos. Por otro lado, la organización depende de la plataforma para que el enlace de comunicación mantenga los datos cifrados, autenticados por la organización. De esta manera la plataforma verifica la identidad con el fin de que los datos en los estándares establecidos sean recibidos para su procesamiento con disponibilidad total.

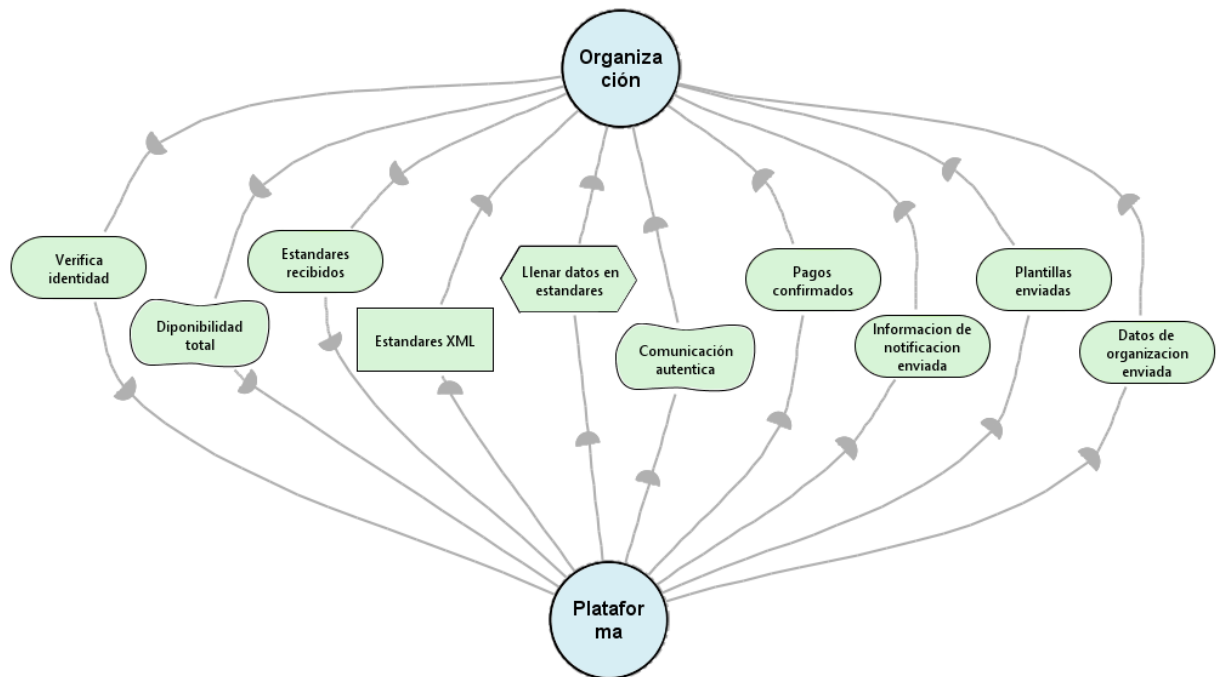


Figura 5-1: Dependencia Plataforma – Organización.

Dependencia entre plataforma y servidor de correo

El servidor de correo depende de la plataforma en la espera de envíos de notificaciones de correo electrónico hacia los clientes. El servidor de correo requiere que la plataforma realice las peticiones de envío y que éstas sean hechas utilizando protocolos para la conexión, como se puede observar en la Figura 5-2.

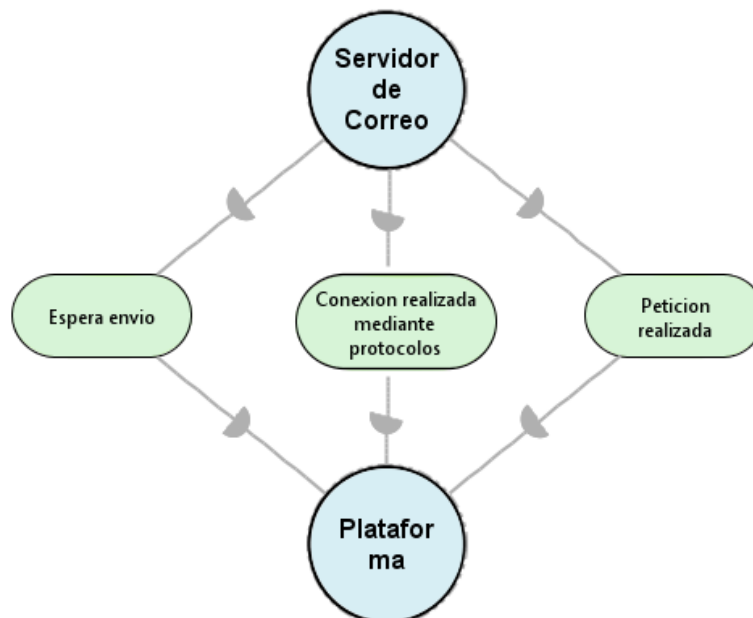


Figura 5-2: Dependencia Plataforma - Servidor de Correo.

Dependencia entre plataforma y pasarela de SMS

La Figura 5-3 nos muestra como la pasarela de SMS depende de la plataforma en la espera de envíos de notificaciones de mensaje de texto hacia los clientes. La pasarela de SMS requiere que la plataforma realice las peticiones de envío y que éstas sean hechas utilizando protocolos para la conexión.

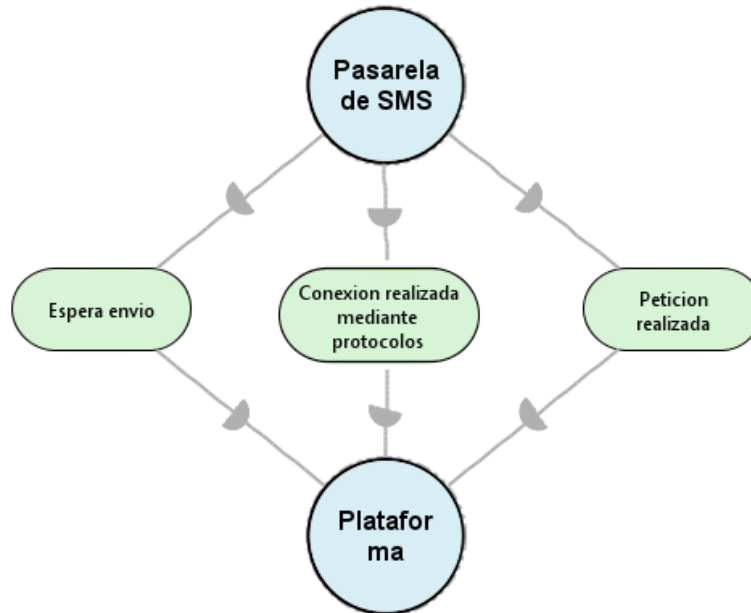


Figura 5-3: Dependencia Plataforma - Pasarela de SMS.

Dependencia entre plataforma y autoridad de certificación

De acuerdo a la Figura 5-4 se puede decir que la plataforma depende de la Autoridad de Certificación para que sea verificada la procedencia de las solicitudes hechas a los servicios web por las organizaciones, utilizando los certificados digitales. La Plataforma requiere que la Autoridad de Certificación sea una entidad segura a nivel de cadena de certificación.

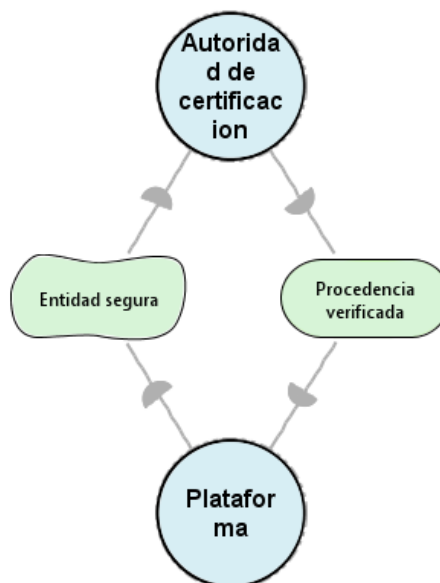


Figura 5-4: Dependencia Plataforma - Autoridad de certificación.

Dependencia entre pasarela de SMS y cliente

En la Figura 5-5 se puede observar como el Cliente depende de la Pasarela de SMS para que ésta realice un envío seguro de las notificaciones, en el despacho de mensajes de texto.

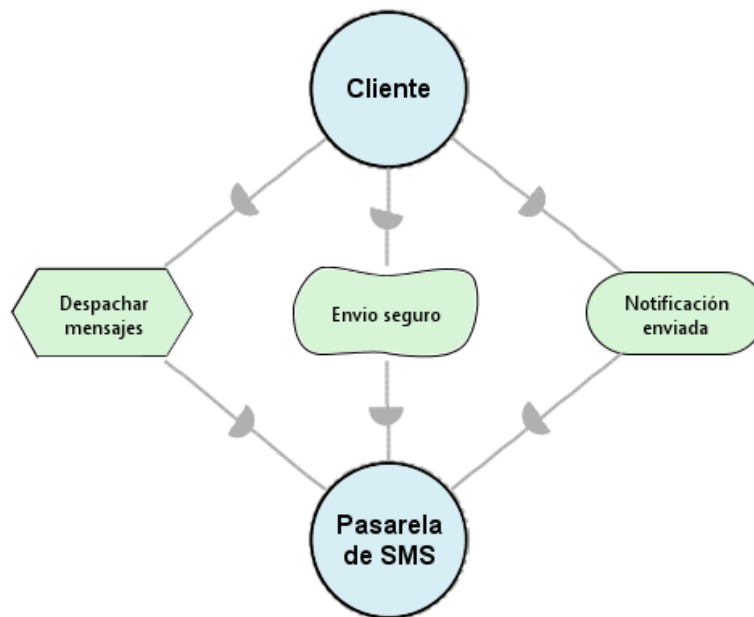


Figura 5-5: Dependencia Pasarela de SMS - Cliente.

Dependencia entre servidor de correo y cliente

En la Figura 5-6 se muestra que el Cliente depende del Servidor de Correo para que éste realice un envío seguro de las notificaciones, en el despacho de correos electrónicos.

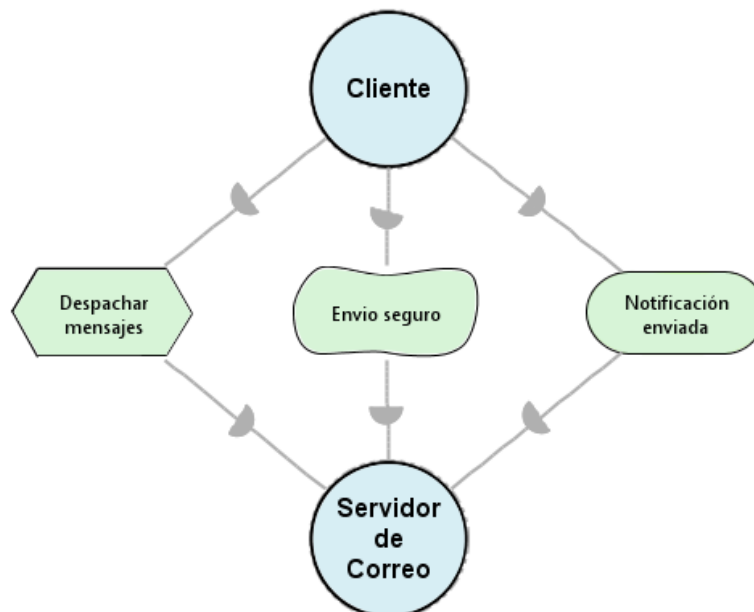


Figura 5-6: Dependencia Servidor de Correo - Cliente.

Diagrama completo SD

A continuación, en la Figura 5-7 se muestra el diagrama completo de dependencias estratégicas. En donde se encuentran los actores y sus dependencias hacia los demás.

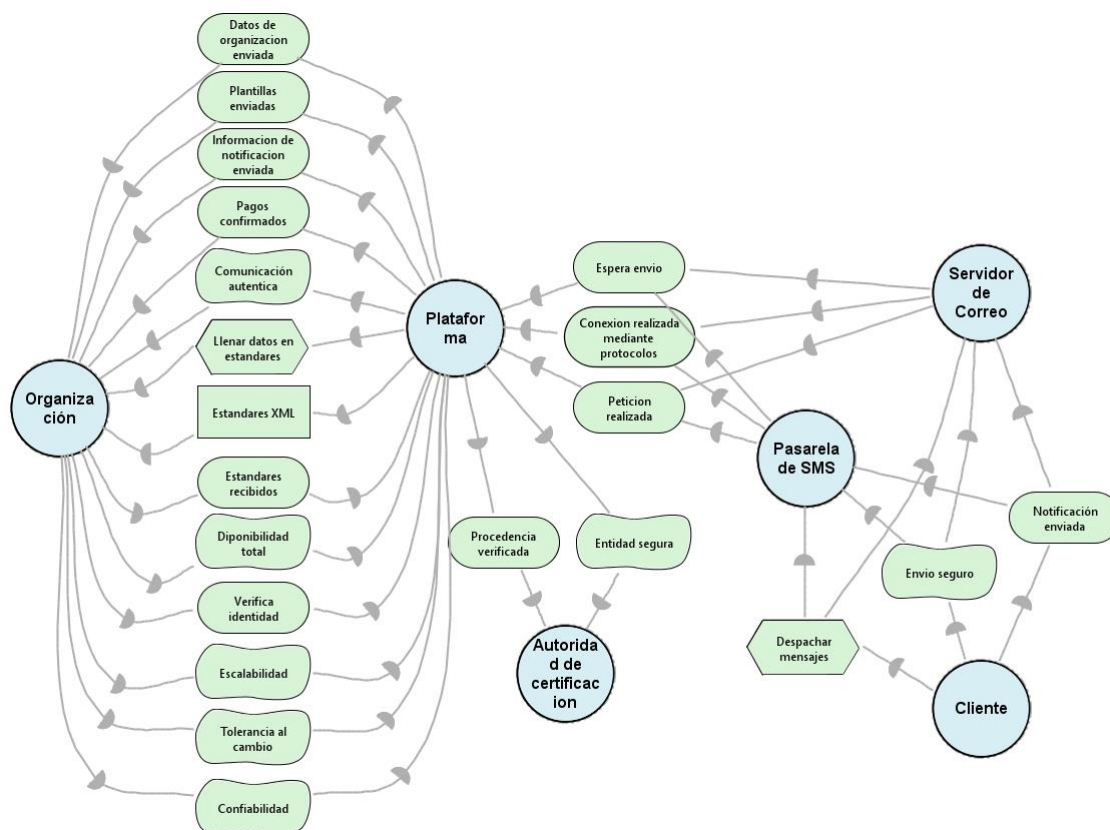


Figura 5-7: Diagrama general SD.

5.2 Fase de análisis y requerimientos

En la fase de análisis y requerimientos se pretende especificar de manera clara y precisa las diferentes características de la plataforma, con el fin de refinar las funcionalidades requeridas. De esta manera se realizan los diagramas racional estratégico (SR – Strategic Rational), se plantean los diversos escenarios, casos de uso, diagramas de secuencia, estados y la arquitectura que tendrá la plataforma.

5.2.1 Descripción de modelo SR, I-Estrella

A continuación, se muestra las descripciones de las diferentes dependencias entre los actores definidos previamente, junto con su modelo respectivo.

Dependencia entre plataforma y organización

La organización realiza varios objetivos y tareas para poder usar el servicio de notificaciones correctamente. Como se muestra en la [Figura 5-8: Diagrama SR Plataforma – Organización](#). Figura 5-8, en primer lugar la organización dispone de los diferentes modelos establecidos en *XML*, con el fin de llenar los diferentes campos. En el caso de requerir enviar la información de la empresa junto con las plantillas para las notificaciones se debe rellenar el estándar *XML* respectivo con dicha información. De la misma manera si desea enviar las notificaciones o las confirmaciones de pagos, la organización debe llenar los modelos *XML* respectivos y firmarlos electrónicamente, con el fin de autenticar dicho los datos. De esta manera la plataforma, brindando una disponibilidad total, verifica la identidad de la organización mediante la comprobación de la firma electrónica y el certificado de autenticidad. Recibiendo así los datos y almacenándolos en la base de datos.

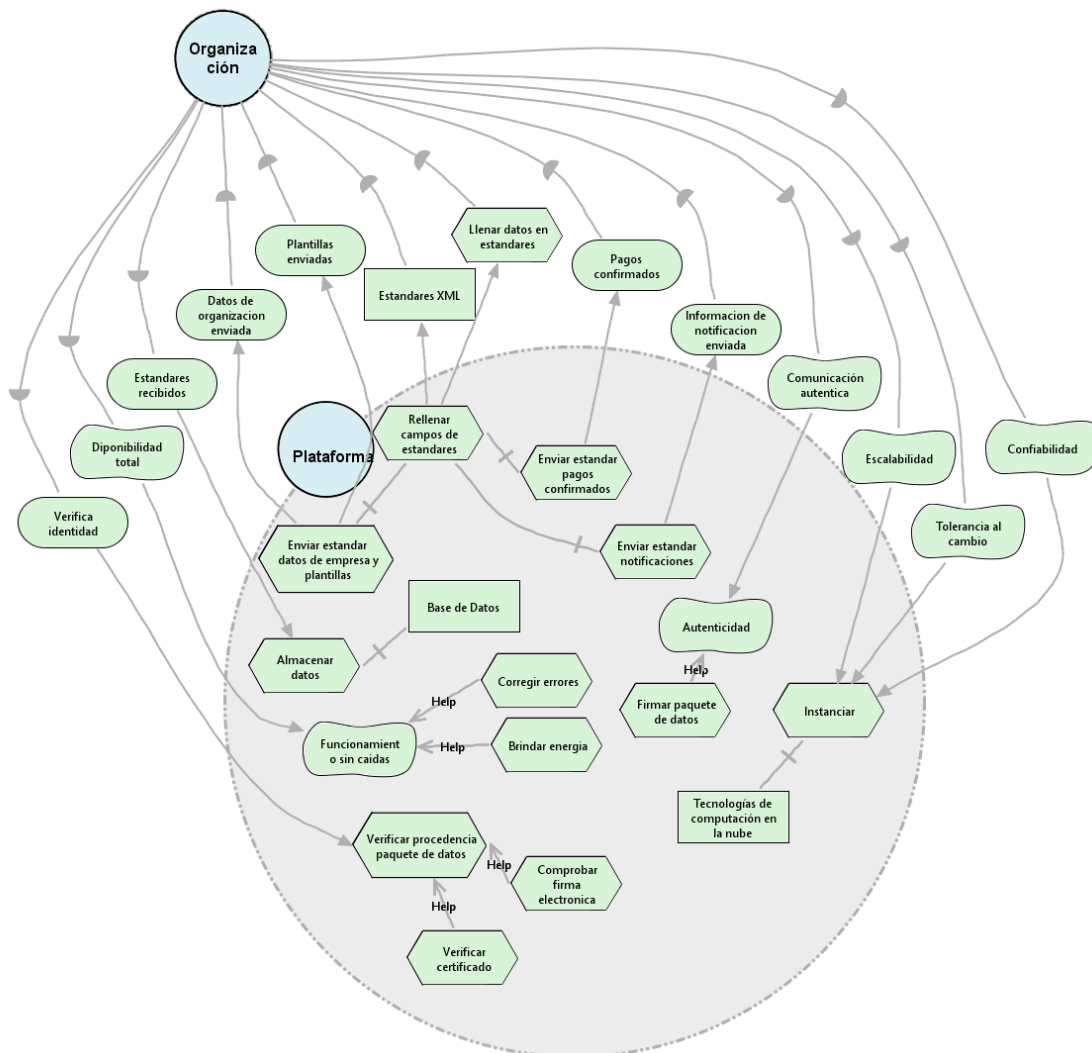


Figura 5-8: Diagrama SR Plataforma – Organización.

Dependencia entre plataforma y servidor de correo

Para las notificaciones mediante correo electrónico es preciso utilizar un servidor de correo como se muestra en la Figura 5-9, de esta manera la plataforma realiza una petición para el envío del mensaje, posterior a esto se realiza una conexión mediante protocolos para la verificación de firmas digitales, con el fin de que el mensaje llegue íntegro al cliente.

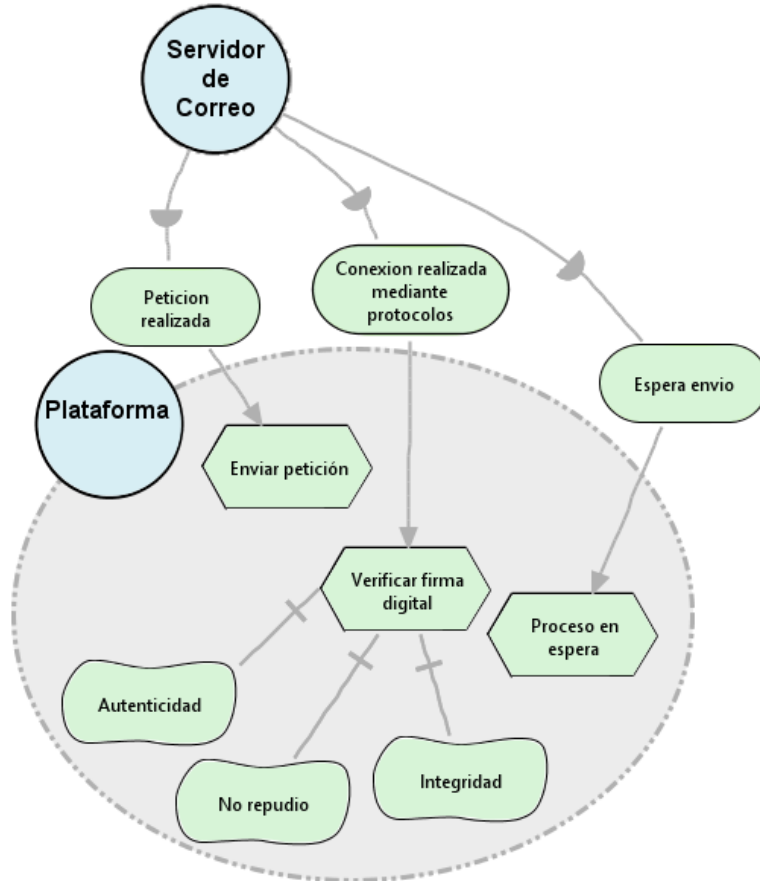


Figura 5-9: Diagrama SR Plataforma - Servidor de Correo.

Dependencia entre plataforma y pasarela de SMS

Para las notificaciones mediante mensajes de texto es necesario recurrir a una pasarela de SMS, como se muestra en la Figura 5-10, de esta manera la plataforma realiza una petición para el envío del mensaje de texto, posterior a esto se realiza una conexión mediante protocolos para la verificación de firmas digitales, con el fin de que el mensaje llegue íntegro al cliente.

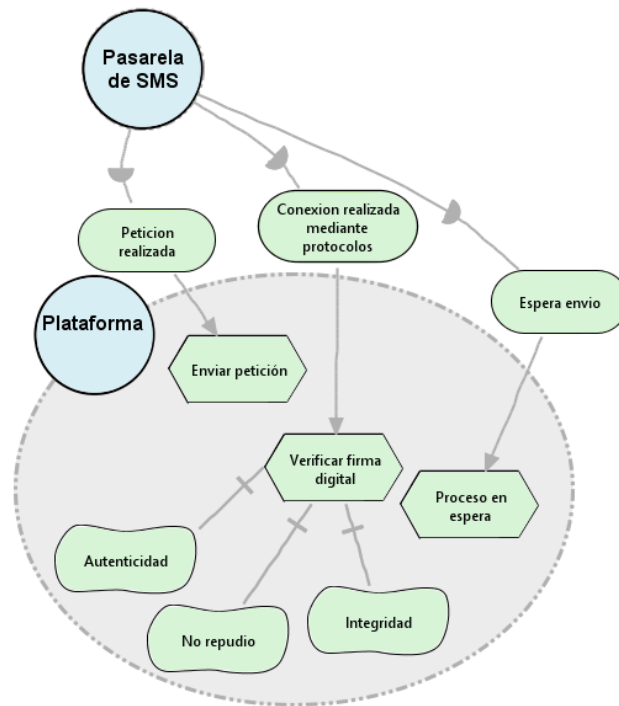


Figura 5-10: Diagrama SR Plataforma - Pasarela de SMS.

Dependencia entre plataforma y autoridad de certificación

Como se observa en la Figura 5-11 la autoridad de certificación, una entidad segura, en este caso el Banco Central del Ecuador verifica la procedencia de la organización mediante el certificado digital para proceder con el proceso respectivo.

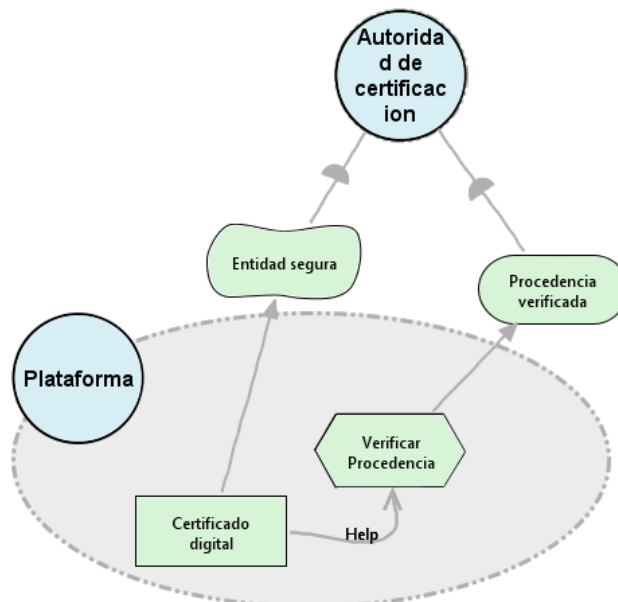


Figura 5-11: Diagrama SR Plataforma – Autoridad de certificación.

Dependencia entre cliente y pasarela de SMS

Para el envío de notificaciones mediante mensaje de texto, se procesa a usar una pasarela de SMS como se muestra en la Figura 5-12. Para que la notificación sea enviada y despachada, por medio de dicha pasarela se realiza un envío seguro, con los datos cifrados. De esta manera se obtiene que las notificaciones lleguen íntegras hacia los diferentes clientes.

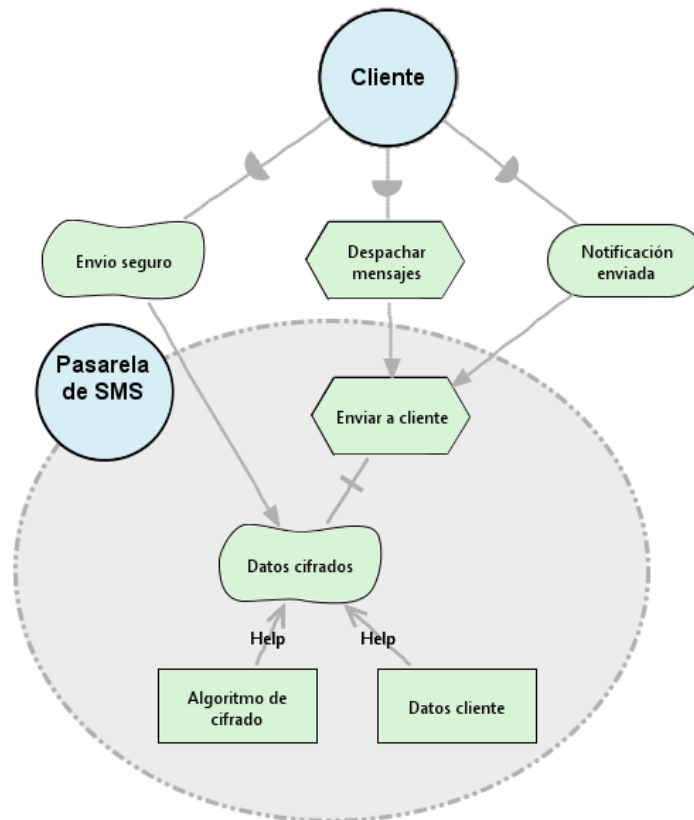


Figura 5-12: Diagrama SR Cliente – Pasarela de SMS.

Dependencia entre cliente y servidor de correo

Para el envío de correo electrónico como notificaciones hacia los clientes de las diferentes organizaciones, se implementa el uso de un servidor de correo electrónico. Así, la Figura 5-13 muestra como el correo se envía de una manera íntegra y segura mediante el cifrado respectivo de dicho servidor.

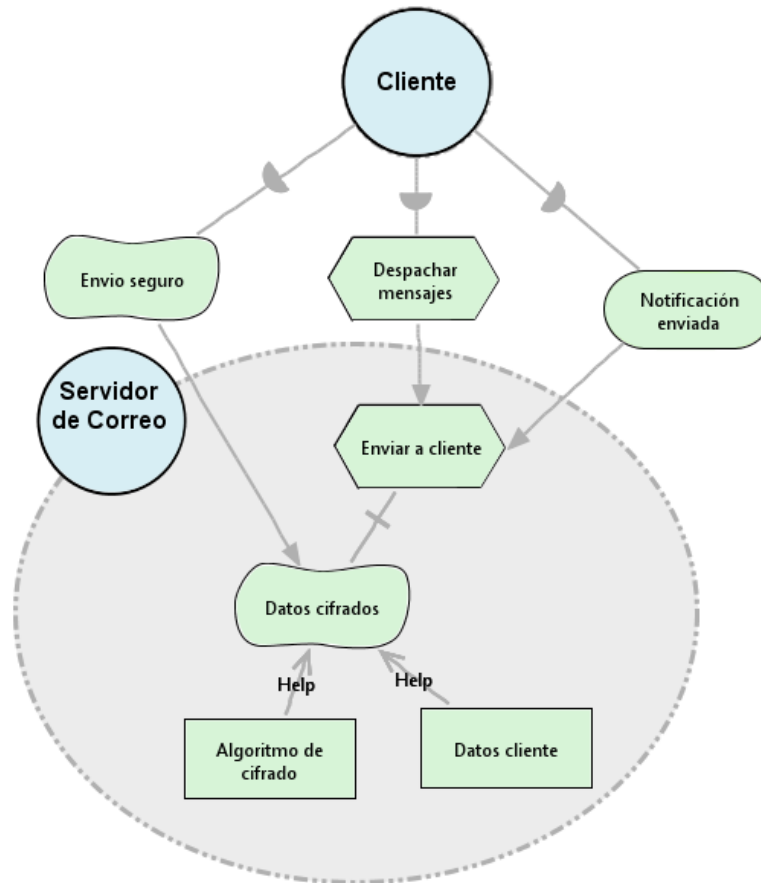


Figura 5-13: Diagrama SR Cliente - Servidor de Correo.

5.2.2 Escenarios

A continuación se presentan los diferentes escenarios que tendrá la plataforma, con el fin de tener claro las diferentes circunstancias que se pudieran dar. En adelante se representaran los escenarios mediante la notación *BPMN*, una descripción detallada de los componentes de dichos diagramas se encuentran en el Anexo B.

Registro y actualización de datos de la organización

De acuerdo como se muestra en la Figura 5-14 la organización genera un archivo *XML* con los datos de la organización en formato *XML* conforme el estándar establecido y lo envía hacia la plataforma. En la plataforma se realiza una verificación de seguridad verificando la autenticidad y validez del certificado de la organización que realiza la petición a los servicios web, luego se realiza el procesamiento de los datos recibidos. Si es la primera vez que la plataforma recibe los datos de la organización procede a crear el perfil, plantilla de correo y plantilla de *SMS* de la organización y lo almacena en la base de datos. Una vez completado este proceso la organización recibe la respuesta de la plataforma culminando el registro de datos. Si la plataforma ya ha recibido los datos de la organización anteriormente, procede a actualizar el perfil, plantilla de correo y plantilla de *SMS* de la organización con los nuevos datos y guarda los cambios en la base de datos. Una vez completado este proceso la organización recibe la respuesta de la plataforma culminando la actualización de datos.

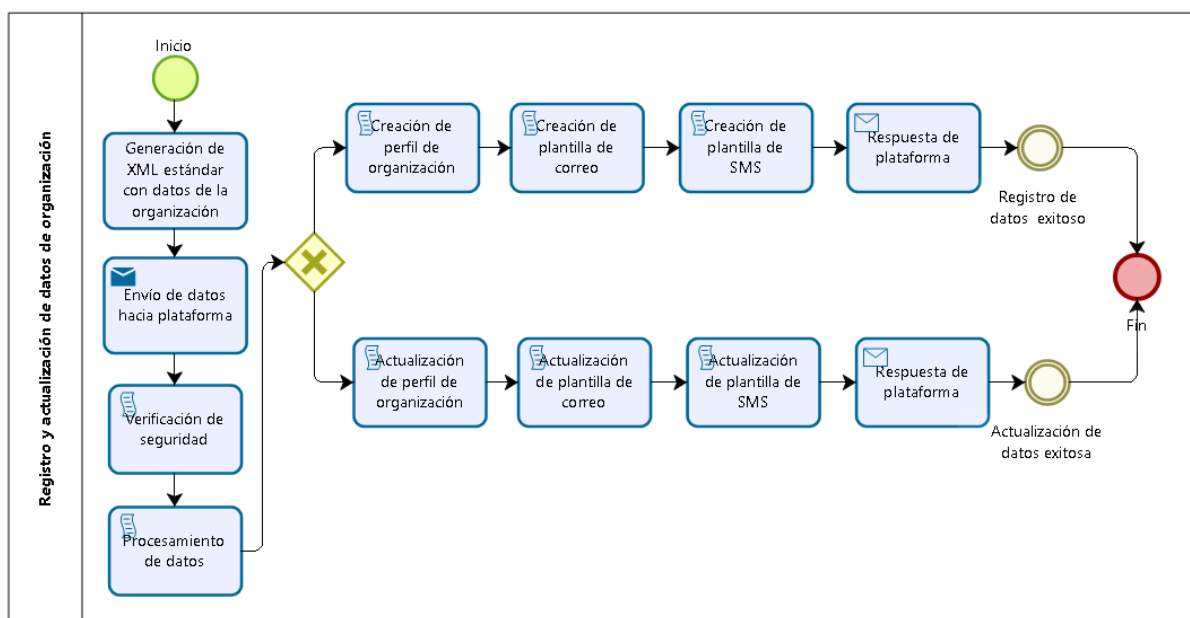


Figura 5-14: Diagrama BPMN del registro y actualización de datos de la organización.

Recepción de comprobantes para notificación

Como se puede observar en la Figura 5-15 la organización genera un archivo *XML* con los datos de los comprobantes que desea modificar, en formato *XML* conforme el estándar establecido y lo envía hacia la plataforma. En la plataforma se realiza una verificación de seguridad verificando la autenticidad y validez del certificado de la organización que realiza la petición a los servicios web, luego se realiza el procesamiento de los datos recibidos. La plataforma procede a validar los comprobantes existentes en el *XML* y registra los comprobantes que no hayan sido recibidos anteriormente, guardando en la base de datos la información correspondiente a los pagos de los comprobantes. Finalmente, la organización recibe la respuesta de la plataforma culminando el proceso de recepción de comprobantes.

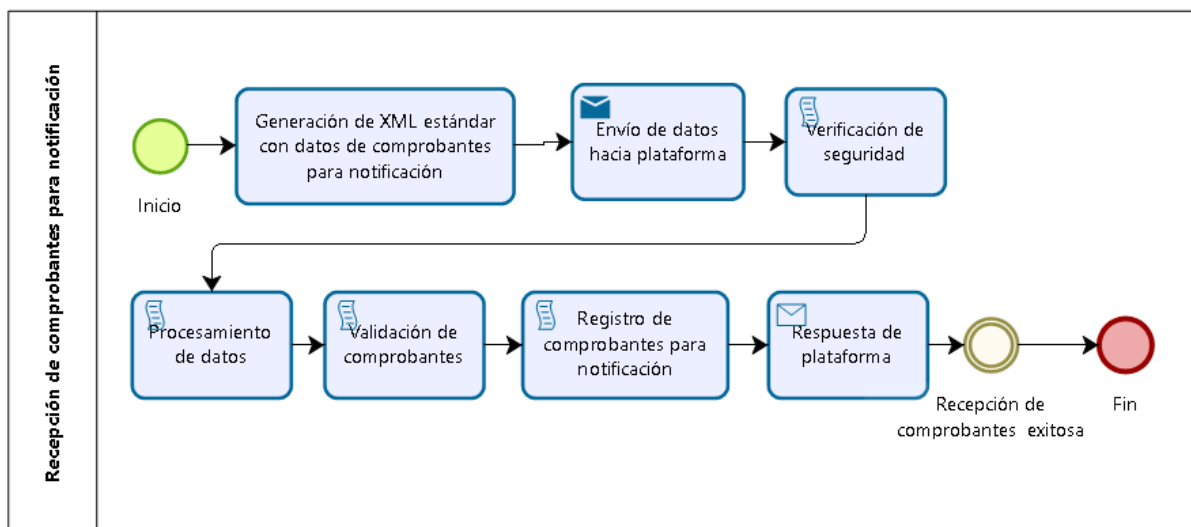


Figura 5-15: Diagrama BPMN de la recepción de comprobantes para notificación.

Confirmación de comprobantes pagados

En cuanto a la confirmación de pagos se observa en la Figura 5-16 como la organización genera un archivo *XML* con los datos de los comprobantes que han sido pagados, en formato *XML* conforme el estándar establecido y lo envía hacia la plataforma. En la plataforma se realiza una verificación de seguridad verificando la autenticidad y validez del certificado de la organización que realiza la petición a los servicios web, luego se realiza el procesamiento de los datos recibidos. La plataforma realiza una validación de los datos de los comprobantes que han sido pagados y desactiva el envío de notificaciones sobre los pagos definidos en el *XML*. Una vez completado este proceso la organización recibe la respuesta de la plataforma culminando la confirmación de comprobantes pagados.

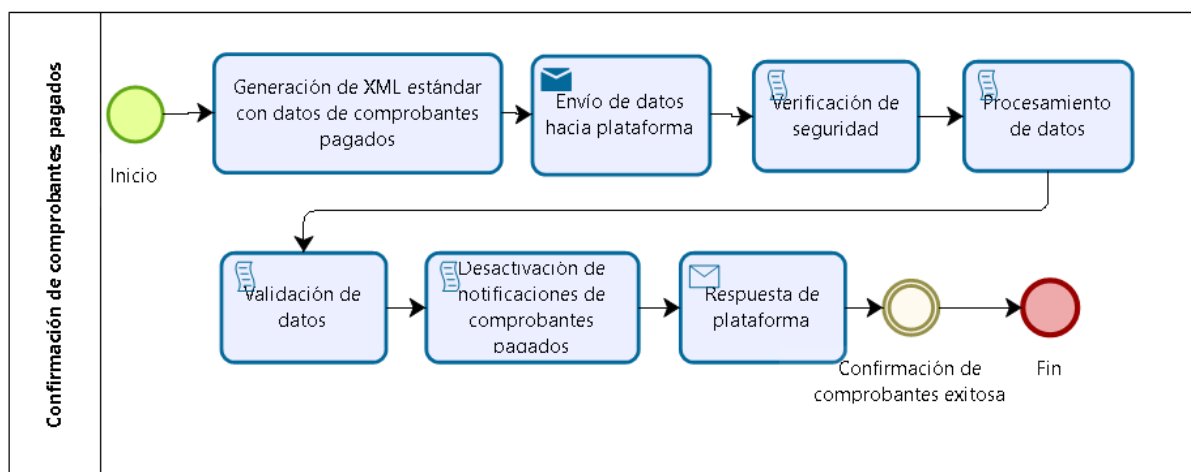


Figura 5-16: Diagrama BPMN de la confirmación de comprobantes pagados.

Envío de notificaciones de correo y SMS

En la Figura 5-17 se observa como la plataforma inicia la ejecución del proceso en segundo plano de notificación conforme las frecuencias establecidas por defecto. Realiza una consulta en la base de datos de los pagos pendientes para notificar, junto con los datos de contacto de sus clientes correspondientes. Si el cliente dispone de dirección de correo electrónico se genera la respectiva notificación de correo electrónico y se la envía hacia el cliente, culminando el envío de notificación de forma exitosa. Si el cliente dispone de teléfono celular se genera la respectiva notificación de SMS y se la envía hacia el cliente, culminando el envío de SMS de forma exitosa.

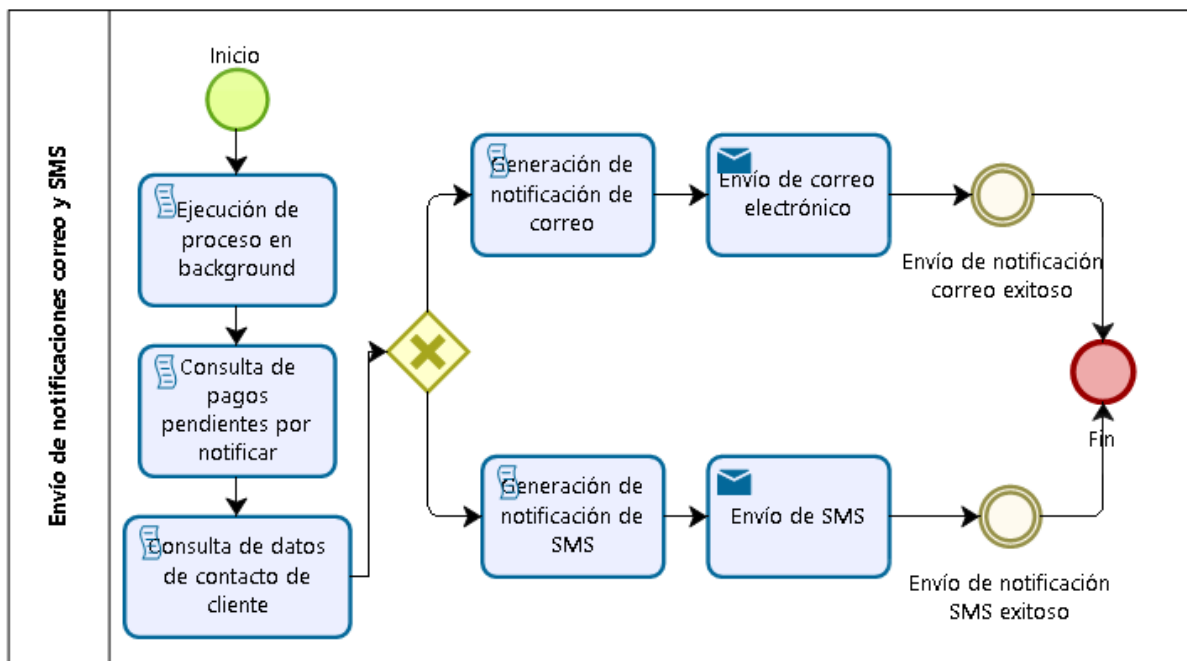


Figura 5-17: Diagrama BPMN del envío de notificaciones de correo y SMS

5.2.3 Casos de uso

Para la descripción de los diferentes pasos o actividades que los actores deben realizar para llevar a cabo algún proceso se tienen descriptos dichos casos de uso en las diferentes tablas a mostrar, Tabla 5-B, Tabla 5-C y Tabla 5-D.

Nombre			Enviar datos y plantillas de empresa.
Actor	Organización, Servidor (Plataforma).		
Descripción	La plataforma deberá comportarse tal como se describe en el siguiente caso de uso cuando una organización envía los datos respectivos en los estándares.		
Precondición	La organización requiere agregar o modificar sus datos de empresa y plantillas (Correo y SMS).		
Secuencia	Paso	Acción	



Normal	1	La organización rellena los campos del estándar XML (datos de empresa y plantillas)
	2	Se envía el estándar respectivamente llenado a la plataforma para su procesamiento.
	3	La plataforma recibe el estándar y almacena tanto los datos de la empresa como las plantillas respectivas para correo y mensajes de texto.
	4	La plataforma confirma la recepción del estándar y el almacenamiento correcto de los datos de la organización.
Postcondición	La organización se encuentra registrada.	

Tabla 5-B: Caso de uso para enviar datos y plantillas de empresa.

Nombre		Enviar notificaciones.	
Actor	Organización, Servidor (Plataforma).		
Descripción	La plataforma deberá comportarse tal como se describe en el siguiente caso de uso cuando una organización envía los datos respectivos en los estándares.		
Precondición	La organización requiere notificar a sus clientes sobre facturas vencidas o con su plazo excedido.		
Secuencia Normal	Paso	Acción	
	1	La organización rellena los campos del estándar XML (datos de clientes, facturas y pagos)	
	2	Se envía el estándar respectivamente llenado a la plataforma para su procesamiento.	
	3	La plataforma recibe el estándar, en donde almacena los datos necesarios para las notificaciones.	
	4	La plataforma confirma que el/los pago/s aún no esté cancelado y envía las notificaciones mediante la respectivas pasarelas.	
Postcondición	El cliente recibe las notificaciones respectivas sobre sus pagos.		

Tabla 5-C: Caso de uso para enviar notificaciones.

Nombre		Confirmación de Pago(s)
Actor	Organización, Servidor (Plataforma).	
Descripción	La plataforma deberá comportarse tal como se describe en el siguiente caso de uso cuando una organización envía los datos respectivos en los estándares.	
Precondición	La organización requiere notificar a la plataforma los pagos cancelados	

	Paso	Acción
Secuencia Normal	1	La organización rellena los campos del estándar XML con los datos respectivos de los pagos cancelados.
	2	Se envía el estándar respectivamente llenado a la plataforma para su procesamiento.
	3	La plataforma recibe el estándar, en donde se comprueba los pagos y da de baja las notificaciones respectivas.
Postcondición	El cliente ya no recibirá notificaciones de sus pagos cancelados.	

Tabla 5-D: Caso de uso para la confirmación de Pago(s).

5.2.4 Diagramas de secuencia

En los siguientes diagramas de secuencia se muestran cómo se procede para el registro y actualización de datos de la organización, recepción de los comprobantes para las notificaciones respectivas hacia sus clientes, confirmación de comprobantes pagados y el envío de notificaciones a través de medios digitales, tales como correo electrónico y mensajes de texto.

Registro y actualización de datos de la organización

Para el registro y actualización de datos se prosigue con los siguientes pasos descritos y mostrados en el diagrama representado en la Figura 5-18.

1. La organización genera el XML con los datos de la organización.
2. La organización hace una llamada a la operación de registrar datos de organización de la plataforma.
3. La plataforma solicita a la autoridad de certificación la verificación del certificado asociado a la operación actual.
4. La autoridad de certificación devuelve una respuesta de verificación a la plataforma.
5. La plataforma realiza el procesamiento de los datos recibidos por la organización.
6. La plataforma guarda los datos recibidos en el perfil de la organización.
7. La plataforma guarda los datos recibidos en la plantilla de correo.
8. La plataforma guarda los datos recibidos en la plantilla de SMS.
9. La plataforma devuelve la respuesta de plataforma correspondiente a los resultados de la operación.

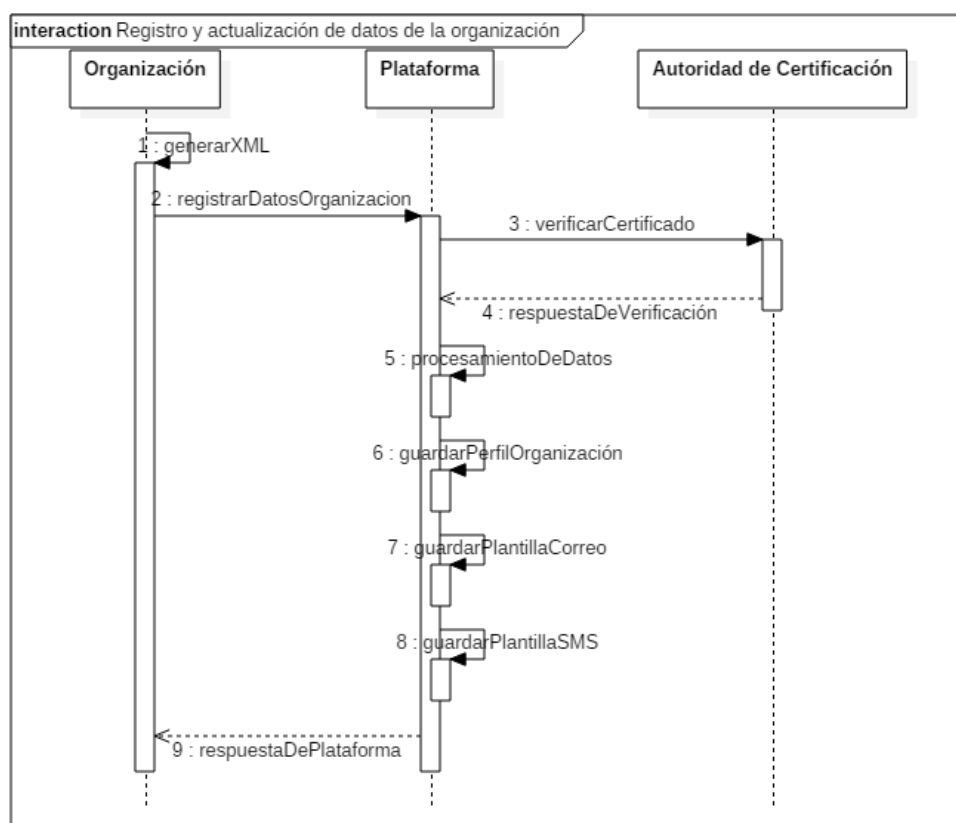


Figura 5-18: Registro y actualización de datos.

Recepción de comprobantes para notificación

Para la recepción de comprobantes para su posterior notificación se prosigue con los siguientes pasos descritos y mostrados en el diagrama representado en la Figura 5-19.

1. La organización genera el *XML* con los datos de los comprobantes que se notificarán.
2. La organización hace una llamada a la operación de recibir comprobantes de la plataforma.
3. La plataforma solicita a la autoridad de certificación la verificación del certificado asociado a la operación actual.
4. La autoridad de certificación devuelve una respuesta de verificación a la plataforma.
5. La plataforma realiza el procesamiento de los datos recibidos por la organización.
6. La plataforma realiza la validación de los datos de comprobantes recibidos.
7. La plataforma realiza el registro de comprobantes para posterior notificación.
8. La plataforma devuelve la respuesta de plataforma correspondiente a los resultados de la operación.

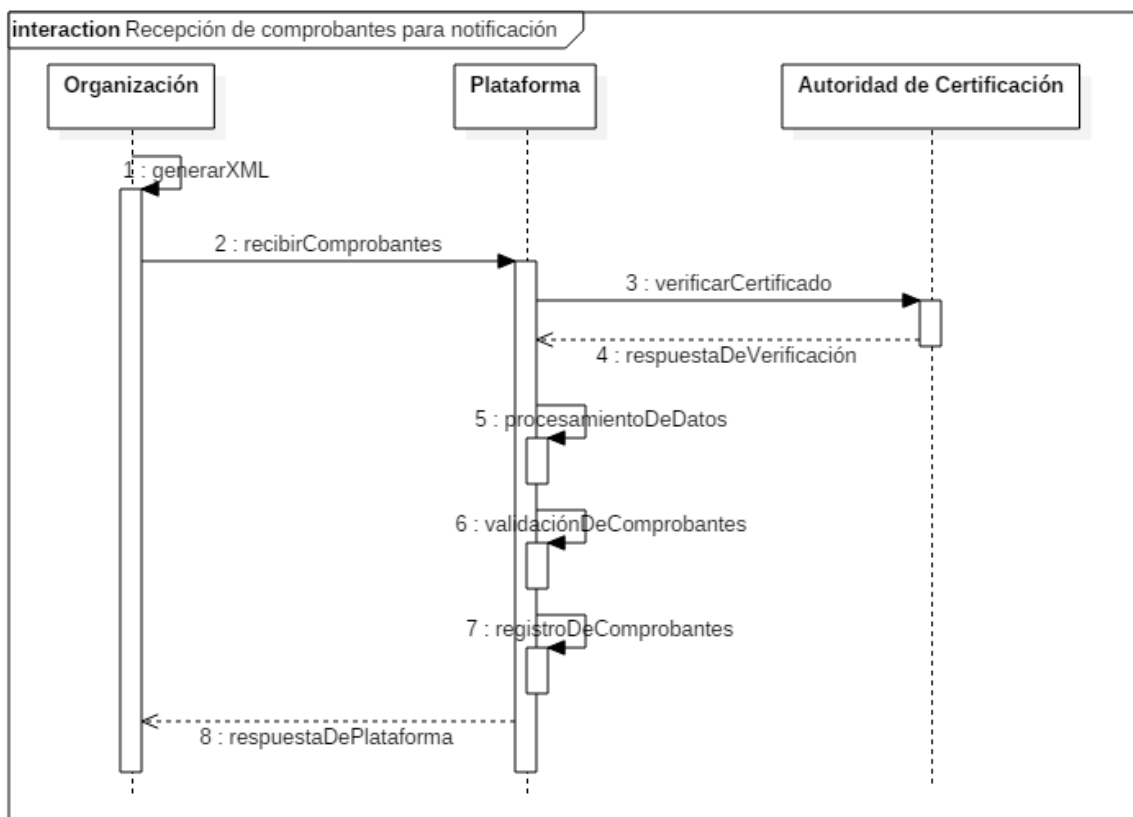


Figura 5-19: Recepción de comprobantes.

Confirmación de comprobantes pagados

Para desactivar las notificaciones mediante confirmación de comprobantes pagados se prosigue con los siguientes pasos descritos y mostrados en el diagrama representado en la Figura 5-20.

1. La organización genera el *XML* con los datos de los pagos de comprobantes a ser confirmados (datos de baja).
2. La organización hace una llamada a la operación de confirmar pagos de la plataforma.
3. La plataforma solicita a la autoridad de certificación la verificación del certificado asociado a la operación actual.
4. La autoridad de certificación devuelve una respuesta de verificación a la plataforma.
5. La plataforma realiza el procesamiento de los datos recibidos por la organización.
6. La plataforma realiza la validación de los datos recibidos por la organización.
7. La plataforma desactiva las notificaciones correspondientes a los pagos definidos en los datos recibidos por la organización.
8. La plataforma devuelve la respuesta de plataforma correspondiente a los resultados de la operación.

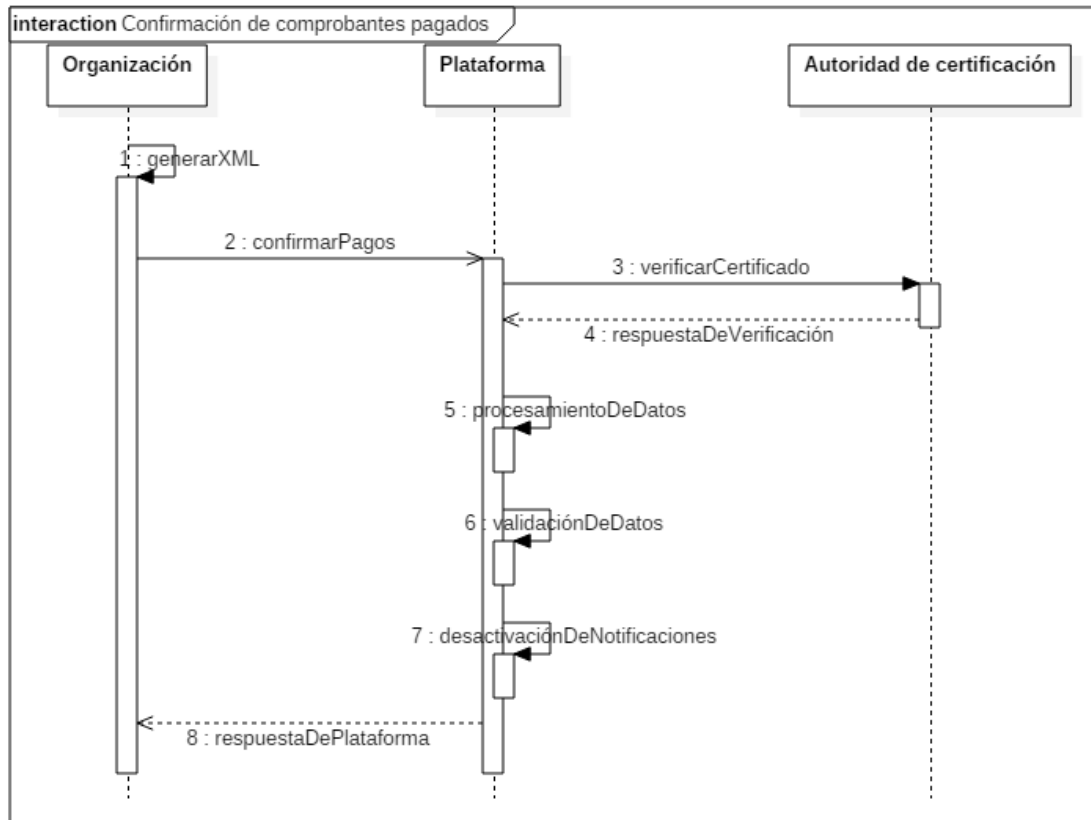


Figura 5-20: Confirmación de comprobantes pagados.

Envío de notificaciones de correo y SMS

Para el envío de notificaciones mediante correo electrónico y mensajes de texto a los clientes se prosigue con los siguientes pasos descritos y mostrados en el diagrama representado en la Figura 5-21.

1. La plataforma inicia la ejecución del proceso de notificaciones en *background*.
2. La plataforma consulta los pagos pendientes para notificación.
3. La plataforma consulta los datos de contacto de los clientes correspondientes a los pagos por notificar.
4. La plataforma solicita el envío de los correos electrónicos utilizando el servidor de correo.
5. El servidor de correo procesa el envío de cada correo electrónico.
6. La plataforma solicita el envío de los mensajes de texto a la pasarela de SMS.
7. La pasarela de SMS procesa el envío de mensajes de texto.

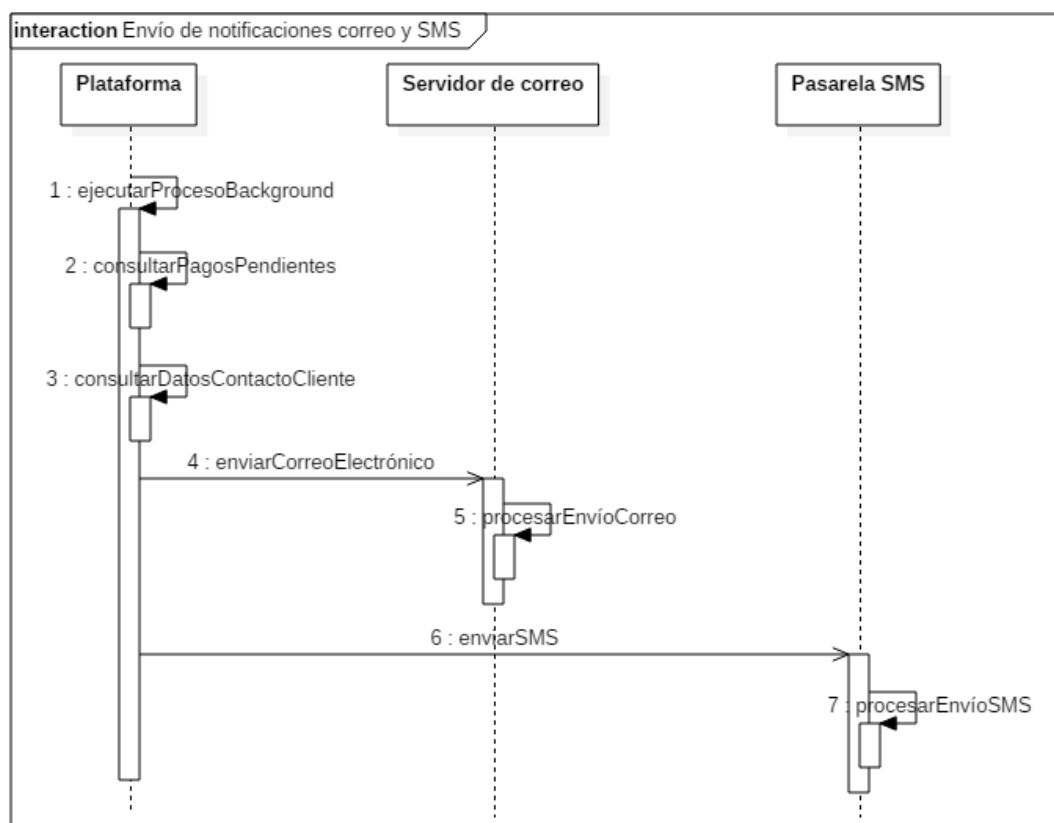


Figura 5-21: Envío de notificaciones.

5.2.5 Diagramas de estado

En los siguientes diagramas de estado se muestran los diferentes estados al momento de que la organización se registre, para el envío de notificaciones y en el caso de confirmar los que se han cancelado los pagos por parte de los clientes.

Registro o actualización de datos de la organización

Para el registro o actualización de los datos de la organización la plataforma se comporta como se muestra en la Figura 5-22.

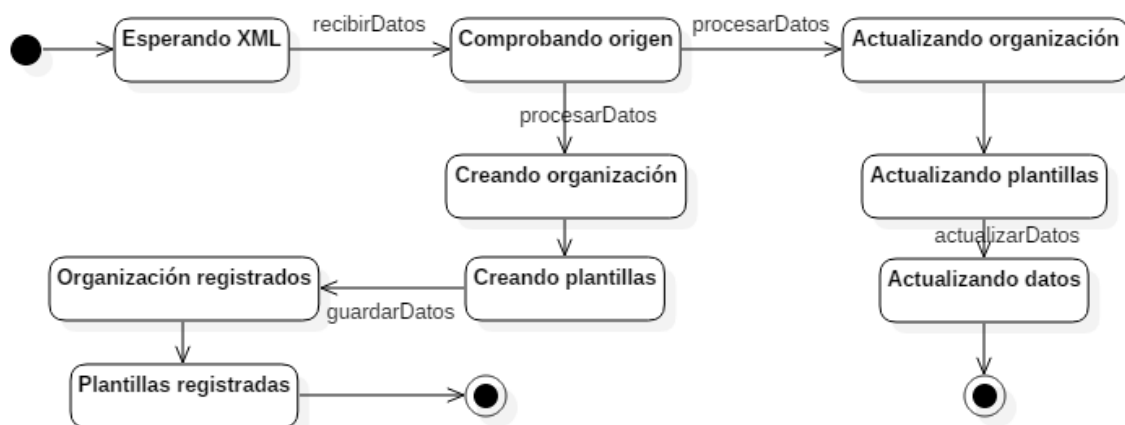


Figura 5-22: Registro o actualización de datos de la organización.

Envío de notificaciones

Para el envío de las notificaciones hacia los clientes de la organización la plataforma se comporta como se muestra en la Figura 5-23.

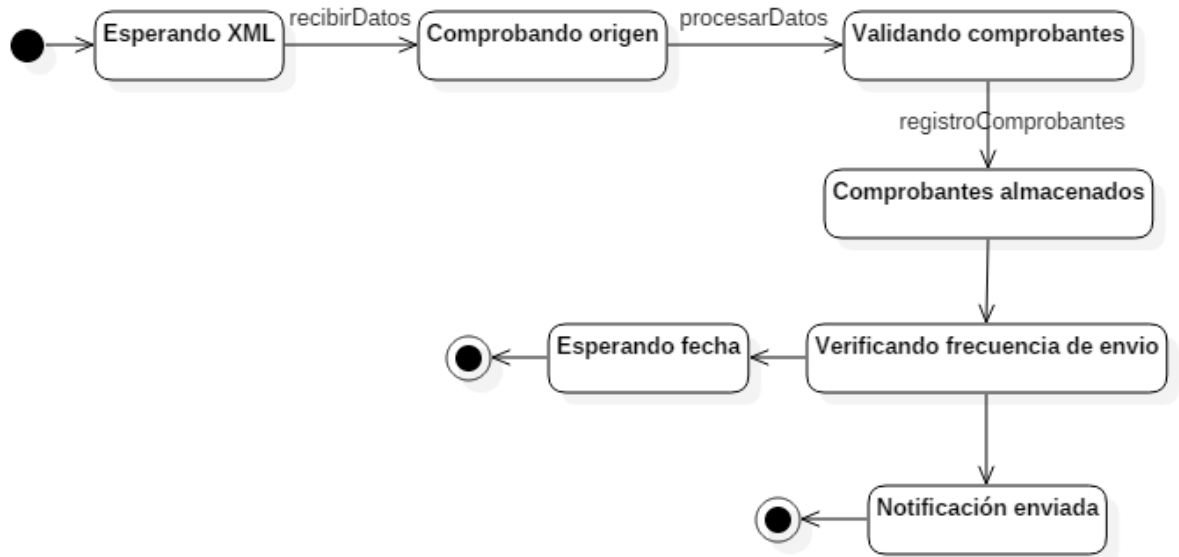


Figura 5-23: Envío de notificaciones.

Confirmación de comprobantes pagados

Para desactivar las notificaciones a un cliente que ya cancelo su pago la plataforma se comporta como se muestra en la Figura 5-24.

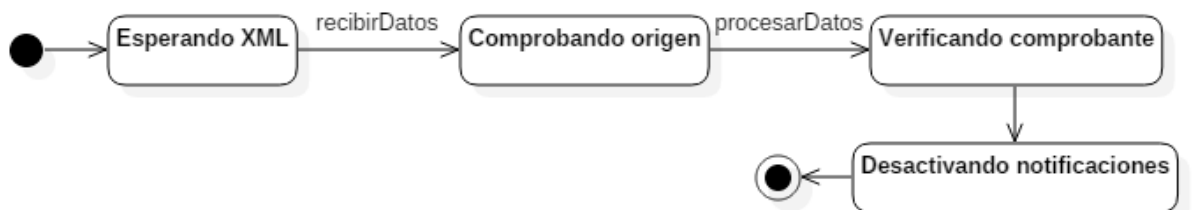


Figura 5-24: Confirmación de comprobantes pagados.



5.2.6 Arquitectura

La arquitectura está estructurada por la plataforma y los elementos que interactúan con ella, de esta manera se considera: computación en la nube, plataforma, organizaciones y comunicaciones de arquitectura.

Computación en la nube

La implementación en la nube se realiza mediante un modelo de despliegue público, que se caracteriza por ofrecer recursos a múltiples clientes o empresas, en este caso, un servicio de notificaciones, permitiéndoles compartir recursos, y por ende realizar importantes ahorros en costos de desarrollo. De esta forma, se proporciona recursos tecnológicos, almacenamiento y software en un modelo de software como servicio (*SaaS – Software as a Service*). De esta forma, el servicio provisto de notificaciones y alertas debe estar completamente disponible, con la posibilidad de crecer y expandir el servicio a más empresas mediante un alto grado de flexibilidad. Asimismo, esta característica es una ventaja cuando hay un desbordamiento del servicio debido a posibles picos de ventas por parte de las empresas en ciertas fechas del año, que al final se convierten en facturas para ser notificadas (Armbrust et al., 2010).

Plataforma

En lo relacionado a la plataforma, lo principal de proporcionar un modelo *SaaS* es cómo se brinda este servicio, ya que es necesario un negocio electrónico para usar tecnologías de *software* que faciliten la conexión de información entre los involucrados (Roy & Ramanujan, 2001). Para esto, se ha elegido el uso de los servicios web, que son una tecnología que utiliza un grupo de protocolos y estándares establecidos por *OASIS* y *W3C*. Gracias al uso de estándares abiertos, es posible tener interoperabilidad entre sistemas de software heterogéneos que cada compañía puede tener (diferentes formatos, plataformas de *hardware* y *software*). Los servicios web responden a solicitudes (*SOAP* - Simple Object Access Protocol), las mismas que están vinculadas mediante la definición de (*WSDL* - Web Services Description Language), esto se basa en un formato universal como *XML* que proporciona varias características como la extensibilidad, la neutralidad y la independencia del *hardware* y los protocolos (Roy & Ramanujan, 2001), también se pueden mencionar otras ventajas del protocolo como una interoperabilidad debido al uso de *XML* y fácilmente escalable mediante el uso de *HTTP*, entre otros.

En cuanto a la base de datos, se ha considerado el uso de *MySQL*, ya que es un sistema de gestión de bases de datos relacionales de tipo *Open Source* y licencia *GPL*. Es uno de los gestores con mejor desempeño para realizar las operaciones actuales, tiene un bajo costo en requerimientos gracias a su bajo consumo de recursos, también tiene una baja probabilidad de corromper datos.

Adicionalmente, se ha implementado el uso de la herramienta (*ORM* - Object-Relational Mapping) de Hibernate. Esto tiene varios beneficios en el momento de la implementación, ya que nos ofrece un paradigma 100% orientado a objetos, de la misma manera que ayuda con un desarrollo ágil y rápido y con muchos menos errores en el tiempo de ejecución, permitiendo hacer un modelado de las entidades mismas (clases de Java) para convertir en las diferentes instrucciones *SQL* (Bauer & King, 2005).



En una vista interna del servidor, se ha considerado una estructura de capas para el procesamiento, para lo cual se ha aplicado el enfoque de *MDA*, que permite la generación de las capas de persistencia y lógica de negocios. De esta forma, se pueden racionalizar los procesos de desarrollo, así como el mantenimiento frente a los cambios en el dominio, logrando la independencia de la plataforma permitiendo la portabilidad del modelo hacia implementaciones específicas de acuerdo con las necesidades específicas de cada proveedor de los servicios web de la plataforma.

Organizaciones

Por parte de las organizaciones, se ha considerado un sistema informático ERP, que debe generar las peticiones en los formatos conforme el estándar *XML* establecido, para enviar notificaciones de pagos, enviar o actualizar datos y plantillas de empresa y confirmar los pagos para que estas notificaciones sean desactivadas. Asimismo, el enlace al *WSDL* debe realizarse para obtener todas las operaciones provistas por los servicios web de la plataforma, el cual estará disponible para el uso de las diferentes organizaciones que solicitan el servicio de notificación.

Comunicaciones de arquitectura

El protocolo utilizado para las comunicaciones entre las organizaciones y el servidor es *SOAP* a través de *HTTP*. El cuerpo de las solicitudes realizadas al servidor consiste en archivos de formato *XML*, que siguen un estándar previamente definido para estructurar la información necesaria para cada operación disponible en el servicio web.

Las solicitudes de servicios web se realizan de forma asíncrona. La plataforma (ii) procesa la información de cada solicitud en una sesión sin estado, cuyo ciclo de vida comienza con la solicitud y finaliza al finalizar el procesamiento. Por otro lado, la plataforma mantiene la ejecución de hilos en paralelo, que están programados para ejecutarse en segundo plano en las frecuencias de notificación establecidas, consultando la base de datos para los pagos cuya notificación está pendiente y enviándola a la pasarela de *SMS* (iii) y servidor de correo (iv).

Las comunicaciones se realizan a través de un canal de comunicaciones seguras utilizando el protocolo (*SSL* - Secure Socket Layer), lo que posibilita el cifrado y la autenticación bidireccional en protocolo de transferencia de hipertexto seguro (*HTTPS* – Hypertext Transfer Protocol Secure), es decir, organización-plataforma y plataforma-organización, utilizando certificados X.509 auto firmados en la plataforma, y otros emitidos por Autoridades de Certificación (*CA* – Certification Authority) para el firmado digital de las peticiones a la plataforma.

Las notificaciones por correo se realizan utilizando el protocolo *SMTP*. Para enviar notificaciones por *SMS* se utiliza una pasarela comercial, que proporciona el servicio de envío de mensajes de texto mediante el uso de solicitudes *HTTP* de tipo *POST* para cada solicitud de envío.

La organización (i) inicia el enlace seguro garantizando el origen de la conexión al servidor verificando el certificado auto firmado en su tienda de confianza. La plataforma (ii) también verifica la validez del certificado de la conexión proveniente, corroborando a través del Protocolo de estado de certificado en línea (*OSCP – Online Certificate Status Protocol*) u otro de funcionalidad similar, con una entidad certificadora de confianza CA (vi). Adicionalmente, se realiza una comprobación de la autenticidad e integridad de los datos recibidos por medio de la firma electrónica adjunta a dichos datos, los cuales deberán seguir el estándar XadES_BES.

Cuando la plataforma (ii) realiza una notificación por *SMS*, realiza una solicitud *POST* a la puerta de enlace *SMS* (iii), incluidas las credenciales de usuario de la puerta de enlace, los datos de destino y el mensaje de texto adjunto. Por otro lado, cuando la plataforma (ii) realiza una notificación por correo electrónico, se utiliza el protocolo *SMTP*, permitiendo que la notificación sea enviada desde el servidor de correo (iv) al cliente destinatario de dicha notificación (v).

Enfoque MDA

En el desarrollo de los servicios web, la tecnología utilizada debe considerarse secundaria, siendo primordiales la información y los servicios disponibles para el tratamiento de dicha información. El uso del enfoque *MDA* permite aumentar la productividad de los desarrolladores al establecer un entorno donde los esfuerzos se pueden concentrar en las tareas de modelado independientemente de las implementaciones en tecnologías específicas.

MDA permite diseñar servicios web a un nivel más abstracto que las implementaciones para tecnologías específicas. Los servicios web desarrollados con *MDA* presentan una mayor flexibilidad y agilidad en el aspecto de los cambios tecnológicos, así como un mayor nivel de calidad y robustez debido a la especificación de dominio más formal y precisa (Frankel & Parodi, 2002).

Por otro lado, *MDA* permite el mantenimiento de sistemas de software a lo largo de su ciclo de vida por los beneficios que conlleva: el mismo *PIM* puede usarse varias veces para generar modelos en diferentes plataformas *PSM*; varias vistas del sistema o varios niveles de abstracción se pueden expresar en el diseño; mejora en la portabilidad e interoperabilidad de los sistemas a nivel de modelo; preservación o evolución uniforme de la lógica comercial frente a cambios o evoluciones tecnológicos; prevención del desarrollo manual propenso a errores; facilidad de migración hacia implementaciones con otras tecnologías (Bézivin, Hammoudi, Lopes, & Jouault, 2004).

Se ha considerado un enfoque de *MDA* para el desarrollo de la plataforma, dadas las características y los beneficios de su uso. Esto permite aplicar el dominio de este trabajo de titulación ajustado de acuerdo a las necesidades específicas de cada organización, independientemente de la plataforma o lenguaje que se utilizará para su implementación, lo que permite la generación automática de código fuente a través de transformaciones.

El dominio de los sistemas de notificación electrónica a los que se aplica este estudio puede estar sujeto a cambios, por lo que el enfoque *MDA* permite ampliar las capacidades de la plataforma con un modelo ya establecido, ahorrando esfuerzo y tiempo.

En este estudio, se ha aplicado una transformación de modelo a texto *M2T*, que permite generar los archivos de código fuente en el lenguaje de programación *JAVA* correspondiente a la capa de negocios y persistencia de la plataforma, basado en el *PIM*, como se muestra en la Figura 5-25.

Estándar XML para consumo de servicios

Los servicios web expuestos por la plataforma procesarán las solicitudes que se realicen desde los sistemas *ERP* de cada organización. Para ello, es necesario definir un estándar que se utilizará para estructurar la información de dichas solicitudes permitiendo el consumo de las operaciones dispuestas en los servicios web: registrar o actualizar la información de perfil de la organización, recepción de comprobantes (facturas) para la notificación, confirmación de pagos.

XML ofrece grandes ventajas en las comunicaciones entre sistemas de software.

Dado que *XML* es un metalenguaje basado en texto, permite crear estructuras que describen información con semántica definida. También es un método independiente de plataforma para describir datos legibles por humanos y máquinas (a diferencia de codificaciones binarias). Dado que se expresa en un formato de texto, es independiente del lenguaje. Además, permite expresar la estructura y las relaciones dentro de los datos utilizando una definición de tipo de documento (*DTD* – Document Type Definition) o un esquema *XML* (*XSD* – XML Schema Definition), lo que permite validar que los datos en *XML* están estructurados correctamente para consumir las operaciones de los servicios web (Chester, 2001).

Teniendo en cuenta estos beneficios, se ha definido un estándar *XML* para cada una de las operaciones mencionadas.

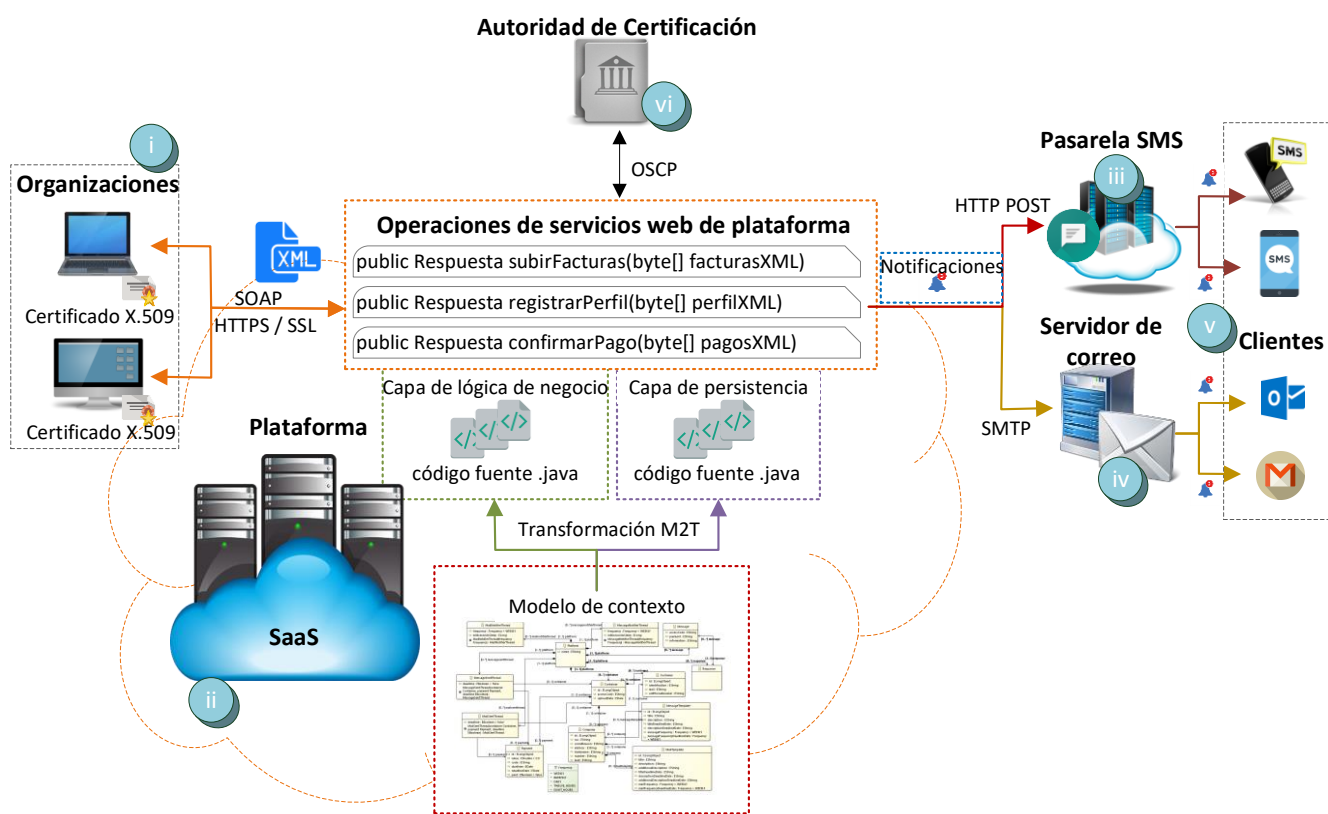


Figura 5-25: Arquitectura de la plataforma.



5.3 Fase de soporte MDA

Una vez realizado todo el análisis de los requerimientos y realizado los diferentes diagramas en donde se muestran los diferentes procesos, se procese con las diferentes actividades que involucra el uso de *MDA*. De esta manera en esta fase se ira desde sus especificaciones del entorno hasta los modelos de contexto necesarios.

5.3.1 Especificar entorno de desarrollo, lenguaje de modelado

Una vez obtenido en fases anteriores los diferentes documentos de requisitos de la plataforma y los requisitos generales, en donde también se tocan los diferentes aspectos tecnológicos del entorno, en el cual la plataforma brindará su servicio.

En esta actividad se procede a especificar a mayor detalle el entorno de desarrollo en el cual se encontrarán la plataforma implementada. Se puede mencionar que en la parte del *hardware*, la plataforma se va a desplegar en una instancia EC2 de tipo T2Medium de *Amazon Web Services*. El mismo que está diseñado para un nivel de desempeño de CPU de referencia con la capacidad de realizar ráfagas para un nivel superior si es que la carga de trabajo lo exige. De esta manera brinda un cierto grado de flexibilidad al momento de su uso (*Amazon Web Services*, 2018). De esta manera no existirán conflictos al momento de que la plataforma se encuentra con sobrecarga de peticiones, siguiendo con sus procesos normalmente. Así mismo, el *software* y las diferentes perspectivas del sistema se encuentran definidas previamente.

Con esto queda especificado el entorno de desarrollo y las diferentes perspectivas del sistema. Adicionalmente se optó por la utilización del lenguaje de modelado a “Java”. Ya que es un lenguaje relativamente sin mucha complejidad con una curva de aprendizaje corta, es un lenguaje orientado a objetos y dispone de una buena seguridad. Así mismo se puede mencionar que es un lenguaje compilado por *bytecode* e interpretado.

5.3.2 Modelos necesarios

Una vez especificado el entorno y seleccionado el lenguaje de modelado se procede a especificar el modelo de contexto que se necesita diseñar para posteriormente realizar las respectivas transformaciones a código e implementar la plataforma de servicios.

De este modo inicialmente, la plataforma va a requerir de un procesamiento paralelo para el envío de notificaciones, tanto de correos electrónicos como mensajes de texto. Así mismo, deberá enviar respuestas a las organizaciones con la información respectiva sobre los diferentes servicios que serán utilizados, con el fin de que la organización esté al tanto de los por menores de sus procesos de notificación. Y así lanzar alertas o avisos en donde se describa cual sea el suceso en el envío de los diferentes datos en el modelo *XML* hacia la plataforma.



Por otro lado la organización podrá indicar sus datos personales, así como ingresar el texto para las diferentes plantillas para las notificaciones, las mismas que tendrán una frecuencia determinada por la organización. Siendo la frecuencia en principio, semanal, bisemanal, diario, cada ocho horas o cada doce horas. Sea para el caso de que las facturas hayan vencido o están con el plazo vencido, esto para los dos tipos de notificaciones que existiría. De la misma manera la organización podrá usar los diferentes servicios gracias a estándares establecidos previamente para los diferentes servicios brindados (envío de notificaciones, envío de datos y plantillas de la organización y confirmaciones de pagos cancelados). En donde existirá un contenedor, el cual incorpora tanto los datos del cliente final como de los pagos de la factura.

5.3.3 Identificaciones y selección de transformaciones y herramientas

Como se mencionó anteriormente para implementar la solución son necesarias las transformaciones de modelo a texto (*M2T*), de esta manera se podrá obtener la capa de persistencia, la capa lógica de negocio y los hilos notificadores (paralelismo). Para lo cual se utilizará *Eclipse Modeling Framework* del IDE Eclipse. Así mismo, se requiere del uso de lenguajes de scripting específicos como *Query/View/Transformation (QVT)* (Kalakota & Robinson, 2001) para tener las diferentes vistas del modelo.

5.3.4 Modelado

Una vez que se tiene toda la documentación en cuanto al proceso del modelado se procede a diseñar dicho modelo. A continuación se muestran los diferentes modelos de contexto para el diseño de la plataforma como se muestra en la Figura 5-26 y Figura 5-27. En donde se puede observar inicialmente en la Figura 5-26 un diagrama en donde es independiente a la plataforma, mientras que en la Figura 5-27 se puede observar un diagrama específico de la plataforma.

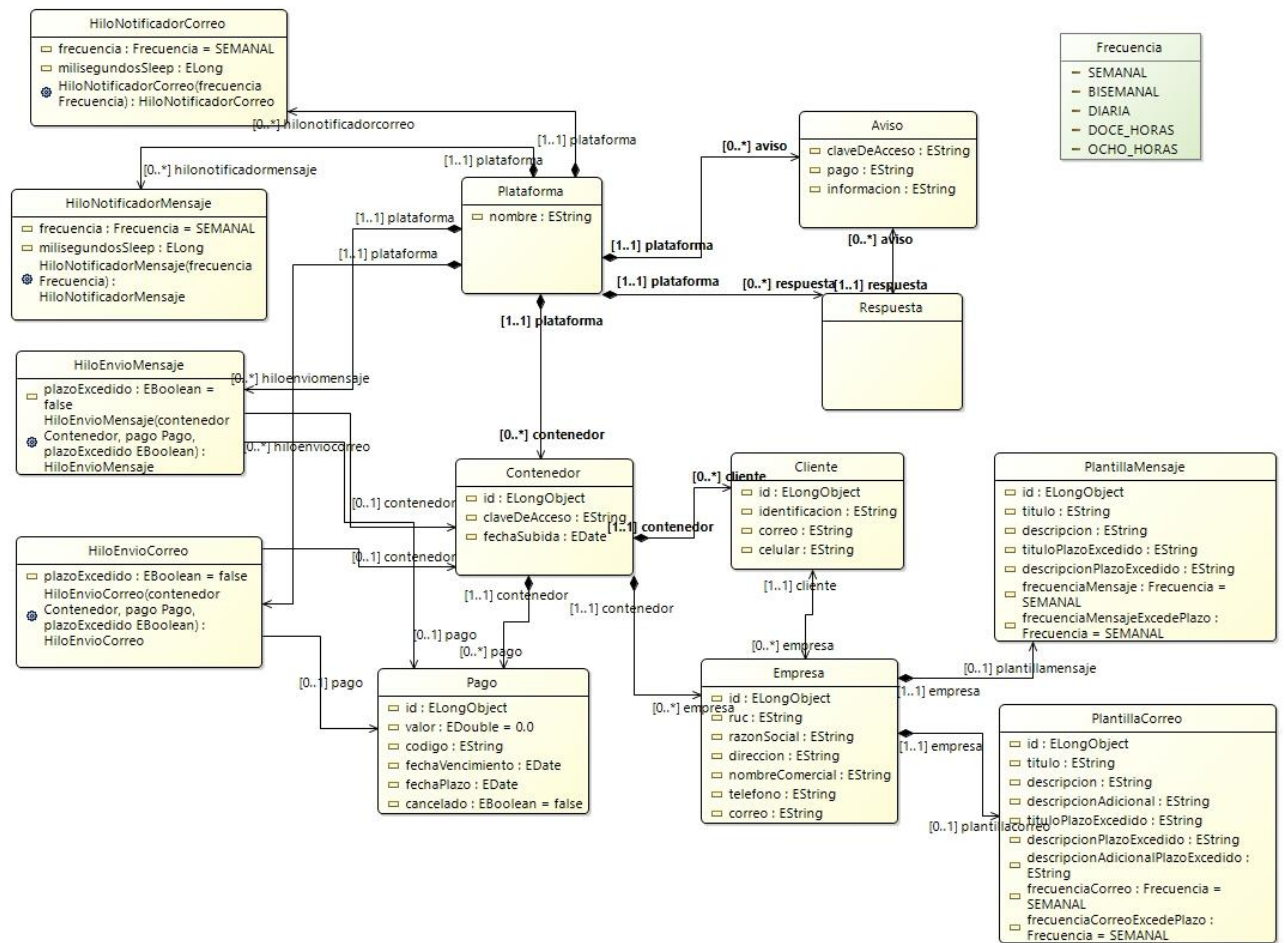


Figura 5-27: Modelo específico de plataforma.

5.4 Fase de desarrollo adaptable

En esta fase se puede apreciar la adopción de la metodología ágil ASD, de esta manera se realizan sus diferentes actividades de especulación, colaboración y adaptación. Aportando de esta manera sus varios beneficios al método propuesto.

5.4.1 Especulación

Basándose en el alcance, conjunto de requisitos de proyecto y de plataforma, modelos de contexto, reglas de transformación, disponibilidad de recursos, se han establecido 4 iteraciones definidas como se muestra en la Tabla 5-E, Tabla 5-F, Tabla 5-G y Tabla 5-H respectivamente:

Iteración 1

Duración	2 semanas
Fecha inicio	4 de octubre de 2017
Fecha fin	18 de octubre de 2017
Objetivo	<ul style="list-style-type: none"> Realizar la transformación modelo a texto, del modelo de contexto definido en la fase de soporte MDA. Esto incluye la generación de código



	<p>fuelle correspondiente a la capa de lógica de negocio y capa de persistencia, en el lenguaje de programación Java.</p> <ul style="list-style-type: none">Definir formato estándar de comunicaciones XML para acceso a las operaciones de los servicios web de la plataforma.
--	---

Tabla 5-E: Iteración 1.

Iteración 2

Duración	3 semanas
Fecha inicio	18 de octubre de 2018
Fecha fin	08 de noviembre de 2018
Objetivo	<p>Integración de código fuente generado en iteración A con implementación de operaciones de servicios web:</p> <ul style="list-style-type: none">Registro y actualización de datos de organizaciónRecepción de comprobantes para notificaciónConfirmación de pagos

Tabla 5-F: Iteración 2.

Iteración 3

Duración	2 semanas
Fecha inicio	8 de noviembre de 2018
Fecha fin	22 de noviembre de 2018
Objetivo	<p>Implementación de proceso en background de notificaciones de correo electrónico y mensajes de texto. Incorporación de mecanismos de seguridad, conexión SSL con certificados digitales X.509, validación con autoridad de certificación Banco Central.</p>

Tabla 5-G: Iteración 3.

Iteración 4

Duración	1 semana
Fecha inicio	22 de noviembre de 2018
Fecha fin	29 de noviembre de 2018
Objetivo	Despliegue de plataforma en servidor de aplicaciones.

Tabla 5-H: Iteración 4.

5.4.2 Colaboración

A continuación, se muestran los resultados obtenidos durante la etapa de colaboración, en el transcurso de las iteraciones definidas en la etapa de especulación.

Iteración 1

En esta iteración se ha realizado la transformación modelo a texto, del modelo de contexto definido en la etapa de soporte *MDA*. Para esto se ha utilizado el lenguaje *MTL*, definiendo cuatro archivos de transformación, que permiten la transformación hacia la capa de lógica de negocio, implementación de *DAO* de la capa de persistencia e hilos notificadores, como se muestra en la Figura 5-28, Figura 5-29 y Figura 5-30 respectivamente.


```

32 [template public generateElement(aPlataforma : Plataforma)]
33 [comment @main/]
34
35 [for (aPlataforma.eAllContents().eClass())]
36 [if (self.name <> 'HiloEnvioCorreo' and self.name <> 'HiloEnvioMensaje' and
37 self.name <> 'HiloNotificadorCorreo' and self.name <> 'HiloNotificadorMensaje')]
38
39 [file ('modelo/'+self.name+'.java', false, 'UTF-8')]
40 @Entity
41 public class [self.name/] implements Serializable{
42 [for (self.eAttributes)]
43 [comment anotaciones/]
44 [if (self.name = 'id')]
45 @Id
46 @GeneratedValue
47 [if]
48 private [getType(self.eAttributeType.name.toString())] [self.name/] ;
49 [/for]
50 [comment atributos de referencias@main/]
51 [if (self.name.toString() = 'Contenedor')]
52 [for (aPlataforma.contenedor.eClass().eReferences)]
53 [if (self.containment)]
54 private [self.name.toUpperFirst()] [self.name/];
55 [/if]
56 [/for]
57 [/if]
58 [if (self.name.toString() = 'Cliente')]
59 [for (aPlataforma.contenedor.cliente.eClass().eReferences)]

```


Figura 5-28: Transformación hacia capa de lógica de negocio.

```

21 [template public generateElement(aPlataforma : Plataforma)]
22 [comment @main/]
23
24
25 [for (aPlataforma.eAllContents().eClass())]
26 [if (self.name <> 'HiloEnvioCorreo' and self.name <> 'HiloEnvioMensaje' and
27 self.name <> 'HiloNotificadorCorreo' and self.name <> 'HiloNotificadorMensaje' and
28 self.name <> 'Aviso' and self.name <> 'Respuesta' and
29 self.name <> 'PlantillaMensaje' and self.name <> 'PlantillaCorreo')]
30
31 [file ('dao/DAO'+self.name+'Impl.java', false, 'UTF-8')]
32
33 package dao;
34
35 import modelo.[self.name.toUpperFirst()];
36 [if (self.name.toString() = 'Cliente')]
37 [for (aPlataforma.contenedor.cliente.eClass().eReferences)]
38 [if (self.lowerBound <> 1 and self.upperBound <> 1)]
39 import modelo.[self.name.toUpperFirst()];
40 [/if]
41 [/for]
42 [/if]
43
44
45 public interface DAO[self.name/]Impl implements DAO[self.name/] {
46
47 [if (self.name = 'Cliente')]
48 [let varClase = String = self.name]

```

Figura 5-29: Transformación hacia implementación DAO, capa de persistencia.



```
20 [template public generateCientificos(Plataforma : Plataforma)]
21 [comment @main/]
22 [for (aPlataforma.eAllContents().eClass())]
23 [if (self.name = 'HiloEnvioCorreo' or self.name = 'HiloEnvioMensaje' or self.name = 'HiloNotificadorCorreo' or self.name = 'HiloNotificador')]
24 [file ('util/'+self.name+'.java', false, 'UTF-8')]
25 public class [self.name/] implements Runnable{
26
27     [for (self.eAttributes)]
28     [if (self.eAttributeType.name = 'Frecuencia')]
29     private [self.eAttributeType.name/] [self.name/] ;
30     [else]
31     private [self.eAttributeType.instanceClassName/] [self.name/] ;
32     [/if]
33 [/for]
34 [for (self.eReferences.name)]
35 [if (self <> 'plataforma')]
36 private [self.toUpperFirst/] [self/] ;
37 [/if]
38 [/for]
39
40
41 [if (self.name = 'HiloEnvioCorreo' or self.name = 'HiloEnvioMensaje')]
42 [if (self.name = 'HiloEnvioCorreo')]
43 public [aPlataforma.hiloenviocorreo.eClass().eOperations.name/][for (aPlataforma.hiloenviocorreo.eClass().eOperations)][for (self.ePar
44 [else]
45 public [aPlataforma.hiloenviomensaje.eClass().eOperations.name/][for (aPlataforma.hiloenviomensaje.eClass().eOperations)][for (self.eP
46 [/if]
47 [for (aPlataforma.hiloenviocorreo.eClass())]
48 [if (self.eAttributeType.name = 'Frecuencia')]
49 [if (self.eAttributeType.name = 'Frecuencia')]
50 [if (self.eAttributeType.name = 'Frecuencia')]
51 [if (self.eAttributeType.name = 'Frecuencia')]
52 [if (self.eAttributeType.name = 'Frecuencia')]
53 [if (self.eAttributeType.name = 'Frecuencia')]
54 [if (self.eAttributeType.name = 'Frecuencia')]
55 [if (self.eAttributeType.name = 'Frecuencia')]
56 [if (self.eAttributeType.name = 'Frecuencia')]
57 [if (self.eAttributeType.name = 'Frecuencia')]
58 [if (self.eAttributeType.name = 'Frecuencia')]
59 [if (self.eAttributeType.name = 'Frecuencia')]
60 [if (self.eAttributeType.name = 'Frecuencia')]
61 [if (self.eAttributeType.name = 'Frecuencia')]
62 [if (self.eAttributeType.name = 'Frecuencia')]
63 [if (self.eAttributeType.name = 'Frecuencia')]
64 [if (self.eAttributeType.name = 'Frecuencia')]
65 [if (self.eAttributeType.name = 'Frecuencia')]
66 [if (self.eAttributeType.name = 'Frecuencia')]
67 [if (self.eAttributeType.name = 'Frecuencia')]
68 [if (self.eAttributeType.name = 'Frecuencia')]
69 [if (self.eAttributeType.name = 'Frecuencia')]
70 [if (self.eAttributeType.name = 'Frecuencia')]
71 [if (self.eAttributeType.name = 'Frecuencia')]
72 [if (self.eAttributeType.name = 'Frecuencia')]
73 [if (self.eAttributeType.name = 'Frecuencia')]
74 [if (self.eAttributeType.name = 'Frecuencia')]
75 [if (self.eAttributeType.name = 'Frecuencia')]
76 [if (self.eAttributeType.name = 'Frecuencia')]
77 [if (self.eAttributeType.name = 'Frecuencia')]
78 [if (self.eAttributeType.name = 'Frecuencia')]
79 [if (self.eAttributeType.name = 'Frecuencia')]
80 [if (self.eAttributeType.name = 'Frecuencia')]
81 [if (self.eAttributeType.name = 'Frecuencia')]
82 [if (self.eAttributeType.name = 'Frecuencia')]
83 [if (self.eAttributeType.name = 'Frecuencia')]
84 [if (self.eAttributeType.name = 'Frecuencia')]
85 [if (self.eAttributeType.name = 'Frecuencia')]
86 [if (self.eAttributeType.name = 'Frecuencia')]
87 [if (self.eAttributeType.name = 'Frecuencia')]
88 [if (self.eAttributeType.name = 'Frecuencia')]
89 [if (self.eAttributeType.name = 'Frecuencia')]
90 [if (self.eAttributeType.name = 'Frecuencia')]
91 [if (self.eAttributeType.name = 'Frecuencia')]
92 [if (self.eAttributeType.name = 'Frecuencia')]
93 [if (self.eAttributeType.name = 'Frecuencia')]
94 [if (self.eAttributeType.name = 'Frecuencia')]
95 [if (self.eAttributeType.name = 'Frecuencia')]
96 [if (self.eAttributeType.name = 'Frecuencia')]
97 [if (self.eAttributeType.name = 'Frecuencia')]
98 [if (self.eAttributeType.name = 'Frecuencia')]
99 [if (self.eAttributeType.name = 'Frecuencia')]
100 [if (self.eAttributeType.name = 'Frecuencia')]
```

Figura 5-30: Transformación hacia hilos notificadores.

La ejecución de las transformaciones antes mostradas genera los archivos de código fuente Java. La Figura 5-31 muestra la capa de lógica de negocio, la cual contiene las clases del dominio definido en el modelo de contexto.

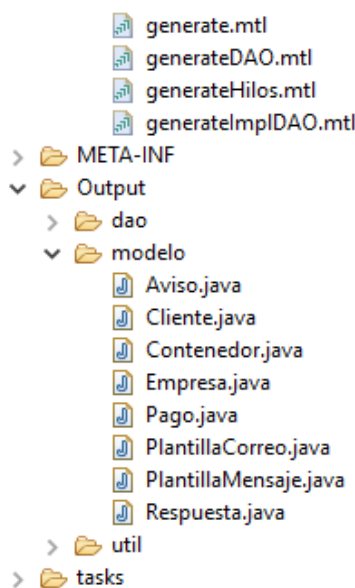


Figura 5-31: Capa lógica de negocio (clases del modelo).

Por otro lado, se ha definido un estándar XML, que consiste en formatos para ser utilizados en el cuerpo de cada petición SOAP realizada a los servicios web de la plataforma, como se muestra a continuación:

Estándar XML para registro y actualización de datos de organización

Para esta operación, el estándar define que el XML contenga la información de contacto de la organización, que consiste en: ruc, razón social, dirección, nombre comercial, teléfono, correo. Adicionalmente deberá contener la plantilla de notificación de correo, es decir: título, descripción, descripción adicional y frecuencia de notificación (SEMANAL, BISEMANAL, DIARIA, DOCE_HORAS, OCHO_HORAS). De igual manera, la plantilla de



notificación de mensajes de texto deberá contener título, descripción, frecuencia de notificación (SEMANAL, BISEMANAL, DIARIA, DOCE_HORAS, OCHO_HORAS).

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="perfilEmpresa">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="infoContacto">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:long" name="ruc"/>
              <xs:element type="xs:string" name="razonSocial"/>
              <xs:element type="xs:string" name="direccion"/>
              <xs:element type="xs:string" name="nombreComercial"/>
              <xs:element type="xs:int" name="telefono"/>
              <xs:element type="xs:string" name="correo"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="plantillaCorreo">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="facturaVencida">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:string" name="titulo"/>
                    <xs:element type="xs:string" name="descripcion"/>
                    <xs:element type="xs:string" name="descripcionAdicional"/>
                    <xs:element type="xs:string"
name="frecuenciaNotificacion"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="facturaPlazoExcedido">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:string" name="titulo"/>
                    <xs:element type="xs:string" name="descripcion"/>
                    <xs:element type="xs:string" name="descripcionAdicional"/>
                    <xs:element type="xs:string"
name="frecuenciaNotificacion"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="plantillaMensaje">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="facturaVencida">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="titulo"/>
              <xs:element type="xs:string" name="descripcion"/>
              <xs:element type="xs:string"
name="frecuenciaNotificacion"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="facturaPlazoExcedido">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="titulo"/>
              <xs:element type="xs:string" name="descripcion"/>
              <xs:element type="xs:string"
name="frecuenciaNotificacion"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```



Estándar XML para recepción de comprobantes para notificación

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="infoRecepcion">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="contenedor" maxOccurs="unbounded" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:string" name="factura"/>
              <xs:element name="cliente">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:int" name="celular"/>
                    <xs:element type="xs:string" name="correo"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="pagos">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="pago" maxOccurs="unbounded"
minOccurs="0">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element type="xs:byte" name="id"/>
                          <xs:element type="xs:float" name="valor"/>
                          <xs:element type="xs:string" name="fechaPlazo"/>
                          <xs:element type="xs:string"
name="fechaVencimiento"/>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Estándar XML para confirmación de pagos

Para esta operación, el estándar define que el XML contenga la clave de acceso del comprobante en cuestión, y los identificadores de los pagos que se desean confirmar sobre dicho comprobante. El formato permite incluir la confirmación de pagos de múltiples comprobantes en una misma solicitud.

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="infoPago">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="contenedor" maxOccurs="unbounded" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:integer" name="factura"/>
              <xs:element name="pagosConfirmados">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:byte" name="pago"
maxOccurs="unbounded" minOccurs="0"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Iteración 2

En esta iteración se ha realizado la integración de código fuente de capa de lógica de negocio y capa de persistencia, generados en la iteración 1, con la implementación de las operaciones de los servicios web de la plataforma:

- Registro y actualización de datos de organización
- Recepción de comprobantes para notificación
- Confirmación de pagos

El documento WSDL muestra las interfaces de las que disponen los servicios web de la plataforma, junto con los tipos de datos necesarios para su utilización.

<definitions>	Este tag es la agrupación de todos los demás elementos.
<pre><definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://ws/" xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:wsp="http://www.w3.org/ns/ws-policy" xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss- wssecurity-utility-1.0.xsd" xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="http://ws/" name="Servicio"></pre>	
<types>	Esta sección define los tipos de datos a utilizar en los mensajes. Para esto se han definido tipos de dato en el XSD.



<pre><types> <xsd:schema> <xsd:import namespace="http://ws/" schemaLocation="http://localhost:8080/Servicio/Servicio?xsd=1" /> </xsd:schema> </types></pre>	
<message>	Consiste en los métodos y parámetros para el uso de las operaciones.
<pre><message name="recibirComprobante"> <part name="parameters" element="tns:recibirComprobante" /> </message> <message name="recibirComprobanteResponse"> <part name="parameters" element="tns:recibirComprobanteResponse" /> </message> <message name="actualizarDatosEmpresa"> <part name="parameters" element="tns:actualizarDatosEmpresa" /> </message> <message name="actualizarDatosEmpresaResponse"> <part name="parameters" element="tns:actualizarDatosEmpresaResponse" /> </message> <message name="confirmarPago"> <part name="parameters" element="tns:confirmarPago" /> </message> <message name="confirmarPagoResponse"> <part name="parameters" element="tns:confirmarPagoResponse" /> </message></pre>	
<portType>	Define las operaciones que se pueden realizar, junto con los mensajes involucrados.
<pre><portType name="Servicio"> <operation name="recibirComprobante"> <input wsam:Action="http://ws/Servicio/recibirComprobanteRequest" message="tns:recibirComprobante" /> <output wsam:Action="http://ws/Servicio/recibirComprobanteResponse" message="tns:recibirComprobanteResponse" /> </operation> <operation name="actualizarDatosEmpresa"> <input wsam:Action="http://ws/Servicio/actualizarDatosEmpresaRequest" message="tns:actualizarDatosEmpresa" /> <output wsam:Action="http://ws/Servicio/actualizarDatosEmpresaResponse" message="tns:actualizarDatosEmpresaResponse" /> </operation> <operation name="confirmarPago"> <input wsam:Action="http://ws/Servicio/confirmarPagoRequest" message="tns:confirmarPago" /> <output wsam:Action="http://ws/Servicio/confirmarPagoResponse" message="tns:confirmarPagoResponse" /> </operation> </portType></pre>	
<binding>	Consiste en la definición del formato de mensaje y detalles de protocolo para cada portType.
<pre><binding name="ServicioPortBinding" type="tns:Servicio"> <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document" /> <operation name="recibirComprobante"> <soap:operation soapAction="" /> <input> <soap:body use="literal" /> </input> </operation> </binding></pre>	



```
</input>
<output>
  <soap:body use="literal" />
</output>
</operation>
<operation name="actualizarDatosEmpresa">
  <soap:operation soapAction="" />
  <input>
    <soap:body use="literal" />
  </input>
  <output>
    <soap:body use="literal" />
  </output>
</operation>
<operation name="confirmarPago">
  <soap:operation soapAction="" />
  <input>
    <soap:body use="literal" />
  </input>
  <output>
    <soap:body use="literal" />
  </output>
</operation>
</binding>
<service name="Servicio">
  <port name="ServicioPort" binding="tns:ServicioPortBinding">
    <soap:address location="http://localhost:8080/Servicio/Servicio" />
  </port>
</service>
</definitions>
```

El XSD describe los tipos de dato con los cuales se utilizarán las operaciones del servicio web.

Esta sección contiene las definiciones de los elementos XML utilizados para las operaciones en los servicios web definidos en el WSDL.

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://ws/" version="1.0" targetNamespace="http://ws/">
  <xs:element name="actualizarDatosEmpresa"
    type="tns:actualizarDatosEmpresa" />
  <xs:element name="actualizarDatosEmpresaResponse"
    type="tns:actualizarDatosEmpresaResponse" />
  <xs:element name="confirmarPago" type="tns:confirmarPago" />
  <xs:element name="confirmarPagoResponse" type="tns:confirmarPagoResponse" />
  <xs:element name="recibirComprobante" type="tns:recibirComprobante" />
  <xs:element name="recibirComprobanteResponse"
    type="tns:recibirComprobanteResponse" />
</xs:schema>
```

Esta sección contiene las declaraciones de cada elemento declarado en la parte superior del XSD, incluyendo nombre, tipo de dato, mínimo de ocurrencias, restricción de nulidad.

```
<xs:complexType name="confirmarPago">
  <xs:sequence>
    <xs:element name="xmlPagos" type="xs:base64Binary" nillable="true"
      minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```



```
</xs:complexType>
<xs:complexType name="confirmarPagoResponse">
  <xs:sequence>
    <xs:element name="return" type="tns:respuesta" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="respuesta">
  <xs:sequence>
    <xs:element name="avisos" type="tns:aviso" nillable="true"
minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="aviso">
  <xs:sequence>
    <xs:element name="informacion" type="xs:string" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="actualizarDatosEmpresa">
  <xs:sequence>
    <xs:element name="datos" type="xs:base64Binary" nillable="true"
minOccurs="0" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="actualizarDatosEmpresaResponse">
  <xs:sequence>
    <xs:element name="return" type="tns:respuesta" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="recibirComprobante">
  <xs:sequence>
    <xs:element name="comprobante" type="xs:base64Binary" nillable="true"
minOccurs="0" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="recibirComprobanteResponse">
  <xs:sequence>
    <xs:element name="return" type="tns:respuesta" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
</xs:schema>
```

Iteración 3

En esta iteración se ha realizado la implementación del proceso encargado del despacho o envío de notificaciones de correo electrónico y mensajes de texto, sobre los resultados de las iteraciones 1 y 2.

Para este propósito se han creado hilos de notificación, los cuales inician de forma automática en el despliegue de la plataforma en el servidor de aplicaciones, utilizando paralelismo para cada uno de éstos de manera que no afecten el procesamiento de las operaciones de los servicios web.

Dado que los requerimientos han establecido la utilización de frecuencias de notificación pre-establecidas (ocho horas, doce horas, diaria, bisemanal, semanal), se ha optado por implementar los hilos de forma que la ejecución de cada hilo pueda ser configurable, permitiendo asignar las frecuencias de notificación correspondientes, logrando que cada hilo realice su proceso de notificación y se detenga por el tiempo establecido para luego reanudar las labores de notificación. Estos hilos se detendrán únicamente al momento de detener la aplicación del servidor de aplicaciones.



De esta manera, el proceso de notificaciones de correo electrónico y *SMS* se da mediante la ejecución de un hilo independiente para cada frecuencia de notificación, resultando en 10 hilos de ejecución:

- Hilo notificador de correo electrónico (configurado a procesar cada 8 horas)
- Hilo notificador de *SMS* (configurado a procesar cada 8 horas)
- Hilo notificador de correo electrónico (configurado a procesar cada 12 horas)
- Hilo notificador de *SMS* (configurado a procesar cada 12 horas)
- Hilo notificador de correo electrónico (configurado a procesar cada 24 horas)
- Hilo notificador de *SMS* (configurado a procesar cada 24 horas)
- Hilo notificador de correo electrónico (configurado a procesar cada 84 horas)
- Hilo notificador de *SMS* (configurado a procesar cada 84 horas)
- Hilo notificador de correo electrónico (configurado a procesar cada 168 horas)
- Hilo notificador de *SMS* (configurado a procesar cada 168 horas)

Durante el tiempo de actividad de procesamiento de cada hilo notificador se consultan los pagos pendientes de cada comprobante, de cada una de las empresas, junto con la información de contacto asociada al cliente de cada pago pendiente, y se procede a realizar el envío de los correos electrónicos y los *SMS*.

Por otro lado, se han incorporado los mecanismos de seguridad de la plataforma, para lo cual se ha utilizado certificados digitales X.509 para establecer enlaces de comunicaciones seguros mediante *SSL*. Adicionalmente, se ha implementado la verificación de certificados con la autoridad de certificación Banco Central del Ecuador.

La autenticación servidor-cliente se ha realizado utilizando un certificado auto firmado de tipo X.509, el cual deberá ser puesto a disposición de cada organización que desee consumir los servicios web provistos por la plataforma, para que puedan configurar sus sistemas ERP haciendo uso de éste certificado. Para la autenticación cliente-servidor, cada organización deberá utilizar el certificado X.509 provisto por la autoridad de certificación Banco Central del Ecuador para configurar sus sistemas ERP y poder acceder a los servicios web de la plataforma mediante el firmado de los datos *XML* que son enviados a las operaciones del servicio web.

En cada invocación de una operación definida en los servicios web de la plataforma se realiza una validación de procedencia de la petición, verificando que los datos de la petición realizada correspondan a la organización propietaria del certificado asociado a esa petición. Por otro lado, se realiza una validación de la validez del certificado en fecha de expiración o revocatoria en la cadena de verificación. En caso de presentarse inconsistencias de propiedad en los datos recibidos y el certificado, la operación será rechazada de inmediato.

Iteración 4

Esta es la iteración que concluye la fase de ejecución y consiste en el despliegue de la plataforma en un servidor de aplicaciones. Para este propósito se ha lanzado una instancia EC2 de Amazon Web Services, que dispone de un sistema operativo *Linux Ubuntu Server 16.04*, con procesamiento AMD de 64 bits con doble núcleo y memoria RAM de 4 gigabytes. Dentro de esta instancia se ha configurado un servidor de aplicaciones *Glassfish* y un servidor de base de datos *MySQL*. La Figura 5-32 muestra detalles relacionados a la instancia en ejecución. Para permitir el acceso público hacia los servicios web de la plataforma se ha abierto el puerto 8080 en donde un servidor de aplicaciones está a la escucha, en este caso es *Glassfish*.



Instance ID	i-0ed218f7824348cdb	Public DNS (IPv4)	ec2-52-32-64-138.us-west-2.compute.amazonaws.com
Instance state	running	IPv4 Public IP	52.32.64.138
Instance type	t2.medium	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-4-104.us-west-2.compute.internal
Availability zone	us-west-2c	Private IPs	172.31.4.104
Security groups	Restrictivo . view inbound rules	Secondary private IPs	
Scheduled events	No scheduled events	VPC ID	vpc-d395a3b7
AMI ID	ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-20171026.1 (ami-0a00ce72)	Subnet ID	subnet-0328955b
Platform	-	Network interfaces	eth0
IAM role	-	Source/dest. check	True
Key pair name	t2medium	T2 Unlimited	Disabled
EBS-optimized	False	Owner	456580185465
Root device type	ebs	Launch time	November 21, 2017 at 12:00:28 PM UTC-5 (2364 hours)
Root device	/dev/sda1	Termination protection	True
		Lifecycle	normal

Figura 5-32: Ejecución de la instancia.

5.4.3 Aprendizaje

En esta sección se han realizado revisiones técnicas a los resultados de las iteraciones de la etapa de ejecución, permitiendo incrementar el conocimiento del equipo de desarrollo y corregir errores o proponer mejoras a los entregables presentados en cada iteración.

Iteración 1

Luego de una revisión técnica de los resultados de la iteración 1 se ha concluido que es necesario modificar las transformaciones modelo a texto, de manera que permitan la generación de elementos de persistencia *DAO* con una implementación independiente de la base de datos. Es decir, al código fuente generado que contiene los *DAOs* como implementación se ha considerado necesario establecer interfaces para cada uno de éstos, permitiendo que cualquier implementación específica de base de datos no tenga acoplamiento con implementaciones de las operaciones de la plataforma. De esta manera, si en algún momento se debería realizar una migración a otra base de datos, simplemente se realizaría una nueva implementación de los *DAOs*, manteniendo las interfaces utilizadas con las operaciones de la plataforma.

En este sentido, se han realizado las transformaciones necesarias para generar los objetos de acceso a persistencia *DAOs*, junto con las interfaces correspondientes para cada uno de estos, como se muestra en la Figura 5-33.

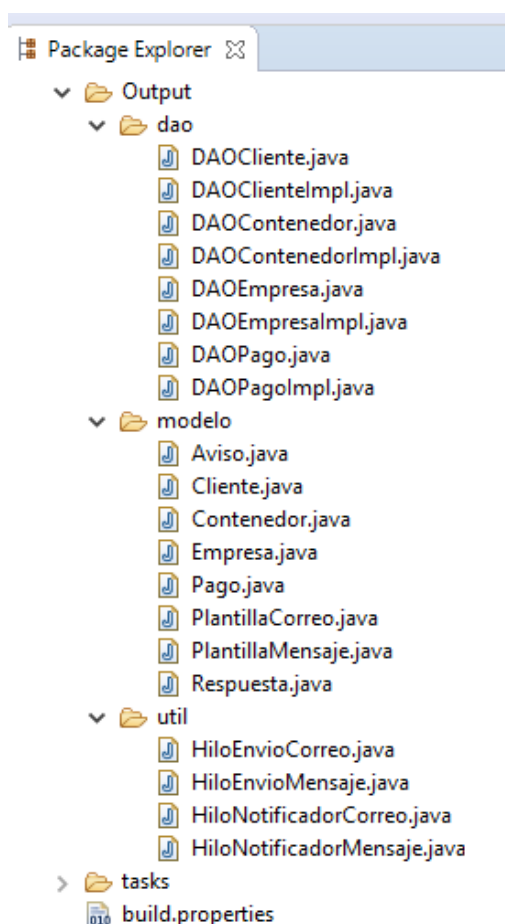


Figura 5-33: Capa de persistencia (Clases).

Iteración 2

Una revisión técnica de la implementación de las operaciones dispuestas en los servicios web de la plataforma ha permitido concluir que las respuestas generadas tras el procesamiento de cada una de las operaciones deben contener información más precisa con respecto a los avisos que se adjuntan. Inicialmente se devolvía una lista de avisos que contenían un atributo de información, incluyendo dentro del mismo el número de pago y la clave de acceso correspondiente, sin embargo esta solución no es la más conveniente por presentar dificultades a las organizaciones para la lectura de las respuestas de la plataforma.

Por lo tanto se ha modificado la respuesta generada por los servicios web en las operaciones, para que incluya en cada aviso un atributo de clave de acceso del comprobante al que corresponde el aviso, y la identificación del pago en cuestión.

```
<xs:complexType name="aviso">
  <xs:sequence>
    <xs:element name="claveDeAcceso" type="xs:string" minOccurs="0" />
    <xs:element name="informacion" type="xs:string" minOccurs="0" />
    <xs:element name="pago" type="xs:string" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

Iteración 3

Una revisión técnica ha permitido determinar que el despacho de notificaciones podría ser optimizado en busca de reducir el tiempo requerido de procesamiento por cada hilo. Dado que el enviar un correo electrónico y un mensaje de texto toma un tiempo



considerable computacionalmente, se ha optado por implementar dos hilos adicionales a los existentes, que realizan las tareas específicas de enviar un correo electrónico y un mensaje de texto, respectivamente, en busca de realizar las labores de despacho de notificaciones de forma rápida. Estos hilos son configurados en cada ejecución con los datos de contacto del cliente a quien le llegará dicha notificación y los datos relacionados al pago del comprobante. Estos hilos utilizan las plantillas de correo electrónico y *SMS* definidas por las empresas, para la redacción del mensaje de la notificación.

Durante el tiempo de actividad de procesamiento de cada hilo notificador se consultan los pagos pendientes de cada comprobante, de cada una de las empresas, junto con la información de contacto asociada al cliente de cada pago pendiente, y se ejecuta un hilo de envío de correo electrónico y un hilo de envío de *SMS* para culminar con el despacho de cada notificación.

Por otro lado, se ha considerado necesario incluir en las notificaciones de correo electrónico información sobre la empresa que emite la notificación y un breve detalle de los elementos de la factura sobre los cuales se está notificando. Esto con el propósito de mejorar la presentación hacia los clientes.

Iteración 4

La revisión técnica de la iteración 4 ha aprobado el correcto funcionamiento de la plataforma, sin embargo, se ha visto necesario incorporar el arranque de los servicios al inicio del sistema operativo, correspondientes al servidor de aplicaciones *Glassfish* y el servidor de base de datos *MySQL*. Esto permite que en casos necesarios o situaciones especiales la instancia pueda reiniciarse y volver a proveer de las funcionalidades de la plataforma de forma automática.

La documentación de requerimientos se ha modificado con cada cambio realizado en las iteraciones, permitiendo que exista concordancia entre las implementaciones realizadas y los requerimientos. Esto permite que cualquier equipo de desarrollo futuro pueda tomar como referencia la documentación de requerimientos para incorporación de nuevas funcionalidades a la plataforma o mantenimientos.



Capítulo 6

Validación mediante caso de uso

El siguiente capítulo tiene como objetivo evaluar mediante un caso de estudio la validez y eficacia de la implementación elaborada mediante el método propuesto.

6.1 Diseño y planeación de caso de estudio

Esta etapa permite la definición clara y documentada de los objetivos, recursos y medios para llevar a cabo el caso de estudio que permita validar el fenómeno en estudio. Para realizar esta validación se ha tomado como referencia las directrices para casos de estudio en ingeniería de software propuestas por Runeson & Höst (2009).

6.1.1 Definir el caso de estudio

Objetivos

En el Capítulo 5 se ha descrito el proceso y los resultados de la instanciación del método propuesto en este trabajo de titulación, constatando la validez específica del desarrollo de la plataforma. Por lo cual, se ha considerado necesaria la validación de dicha implementación en su contexto externo mediante la utilización de un caso de estudio.

De esta manera se pretende validar que la plataforma desarrollada muestra agilidad para las organizaciones. El objetivo de este caso de estudio es validar que la implementación realizada utilizando el método propuesto en este trabajo de titulación, cumple los propósitos planteados en el Capítulo 1, permitiendo la recepción de datos de las organizaciones y el envío de notificaciones hacia sus clientes.

El caso de estudio

El caso de estudio se ha definido como el análisis de la integración y utilización de los servicios provistos por la plataforma en la nube, con los sistemas ERP pertenecientes a la organización GSoft – Soluciones Informáticas, por medio de la construcción de un prototipo.

GSoft – Soluciones Informáticas es una micro-empresa radicada en la ciudad de Cuenca desde el año 2017, dedicada a proporcionar software siguiendo un modelo de *SaaS*. Dentro de la gestión de los procesos de esta organización se incluye la facturación electrónica por servicios brindados; sin embargo, sus servicios no disponían de un mecanismo de notificación adecuada hacia sus clientes, sobre los pagos relacionados a dichos servicios.

Marco teórico

En base a la revisión sistemática que se realizó, se ha mostrado el estado actual de cómo se encuentran este tipo de plataformas que ofrecen un servicio como el propuesto y el uso de tecnologías para ello. En donde se ha visto cómo se han ido adoptando ciertos sistemas de manejo de facturación electrónica, sin embargo, no se encontró una plataforma que brinde un servicio de notificación hacia los clientes finales de las organizaciones, haciendo uso de metodologías como *ASD*, *MDA* y mecanismos de seguridad informática. Bajo esta premisa se podría comprobar la validez del servicio mediante el envío correcto de notificaciones, comprobar que no existan falencias en el manejo de las facturas, pagos y clientes. Adicionalmente se mostrarán los efectos percibidos por el personal encargado de las cobranzas, tras el uso de los servicios provistos por la plataforma en la nube.

Preguntas de investigación

El propósito de este caso de estudio no está basado en términos de preguntas de investigación. Por el contrario, este caso de estudio pretende investigar los efectos



producidos al introducir la utilización de servicios de notificación mediante la plataforma en la nube. Específicamente, con este estudio se busca descubrir posibles debilidades del servicio brindado por la plataforma, así como aspectos positivos que aporten agilidad a la organización en las labores del personal.

Método de recolección de datos

Se ha considerado propicio realizar una entrevista con miembros del personal de la organización, con el fin de conocer el contexto laboral de la organización, los procesos de servicio, y la forma de notificación y gestión de cobranzas que manejan.

Para las entrevistas se ha considerado necesario utilizar un modelo semi-estructurado de entrevista, a manera de discusión permitiendo la exploración y flexibilidad en la información obtenida por este medio.

Por otro lado, es necesario realizar observaciones directas en la organización, con el propósito de recolectar los datos sobre los resultados obtenidos tras la integración y utilización de la plataforma desarrollada conforme el método propuesto, incluyendo los beneficios o debilidades encontradas.

Estrategia de selección

La organización fue seleccionada de la red de contacto de los estudiantes, sobre la base de disponibilidad. De esta manera se trató de escoger un caso en el cual se tenga adoptada la facturación electrónica de forma correcta, como prerequisite de comprobante de venta correspondiente a los servicios ofrecidos por la organización. Es así, como se seleccionó a GSoft – Soluciones Informáticas como el más óptimo para realizar el caso de estudio respectivo.

Consideraciones éticas

La organización GSoft dispone de las siguientes consideraciones para el manejo de los datos de la organización y clientes de la misma.

La organización GSoft aprueba la aplicación del presente caso de estudio, permitiendo la adopción del servicio de notificaciones ofrecido por la plataforma, conforme se establece en el Anexo D.

La organización GSoft ha solicitado la no divulgación de datos de sus clientes, que brindara a los estudiantes, conforme se establece en el Anexo D.

6.2 Reelección de datos

Es necesario conocer cómo la organización se maneja en el fenómeno de interés para este caso de estudio, de forma cualitativa y cuantitativa. Por lo tanto, se ha considerado conveniente aplicar observaciones en la organización y una entrevista semi-estructurada.

Observaciones

Para la recolección de datos se ha visto necesario conocer cómo se están abordando los procesos en la organización, específicamente la notificación de los clientes de la organización por los pagos pendientes respecto a los servicios brindados. Para este propósito se ha planificado una visita a la organización con el fin de obtener los datos preliminares mencionados.

Entrevistas

Para establecer correctamente el contexto de las entrevistas, se han definido los siguientes aspectos:



Evaluar:	Los efectos positivos o negativos percibidos de forma cualitativa y cuantitativa
Con el propósito de:	Validar el método propuesto con respecto a los beneficios encontrados en la organización
Desde el punto de vista de:	Investigadores
En el contexto de:	Personal de cobranzas de la organización

Dada la naturaleza de la información requerida para este caso de estudio, se ha diseñado una entrevista semi-estructurada, permitiendo recolectar datos para un posterior análisis cualitativo y cuantitativo, como se muestra a continuación:

Sección estructurada:

En esta sección se utiliza una escala del 1 al 5 para que el personal de cobranzas valore cada una de las preguntas, como se muestra a continuación:

1. Nada 2. Muy poco 3. Moderadamente 4. Bastante 5. Completamente

¿La utilización de los servicios de la plataforma para la gestión de notificaciones ha facilitado sus labores de cobranzas?

¿La utilización de los servicios de la plataforma para la gestión de notificaciones le ha permitido reducir el tiempo operacional de notificar a sus clientes?

¿En qué medida usted considera que el uso de los servicios de la plataforma para la gestión de notificaciones ha disminuido la tasa de errores?

Sección semi-estructurada

Esta sección de la entrevista permite al personal de cobranzas dar su opinión de forma abierta, permitiendo explorar y descubrir información importante sobre los aspectos a evaluar.

¿Cómo ha sido su experiencia en la gestión de cobranzas utilizando la plataforma para el envío de notificaciones?

¿Qué aspectos le han parecido interesantes o útiles en el proceso de notificaciones?

¿Qué experiencia tuvo en la utilización de la operación de registro de organización?

¿Qué experiencia tuvo en la utilización de la operación de subida de comprobantes para notificación?

¿Qué experiencia tuvo en la utilización de la operación de confirmación de pagos?

Métricas

Las técnicas de recolección de datos utilizadas permiten analizar datos de forma cualitativa y cuantitativa. Sin embargo, para los datos cuantitativos se requiere de medidas

de *software*. La medición de software es el proceso que permite representar las entidades de *software* como procesos, recursos y productos, en números cuantitativos (Fenton, 1999). El propósito de este caso de estudio requiere que se pueda cuantificar la agilidad percibida por la organización tras la utilización de los servicios provistos por la plataforma.

Por lo cual, se considera como unidad de medida el número de días transcurridos desde el envío de la primera notificación de pago, hasta la cancelación de los valores correspondientes al pago en cuestión. De la misma manera, se plantea a los minutos como unidad de medida de tiempo para el análisis de eficiencia en el proceso de notificaciones.

6.3 Análisis de datos recolectados

Para el análisis de datos se han transcrito las entrevistas realizadas al personal de la organización, de forma literal, exceptuando pausas, y otros gestos que se han considerado ajenos a la entrevista, como se muestra en el Anexo E.

En el transcurso del análisis la información toma distintas formas vistas desde diferentes niveles de abstracción, como se muestra a continuación:

1. Los individuos interactúan en su contexto organizacional.
2. Los puntos de vista o percepciones de cada individuo, basadas en sus observaciones y experiencias, son discutidos en entrevistas semi-estructuradas.
3. Las entrevistas son grabadas en archivos de audio.
4. Se transcriben las entrevistas en su totalidad.
5. Se identifican citas o frases interesantes del material transcrito.
6. Las citas o frases encontradas son depuradas y agrupadas para mostrar la relación entre sí.
7. Se identifican resultados individuales de éstos grupos como conclusión.

Cada cambio en la forma de representación de la información es resultado de algún tipo de tratamiento, como se muestra en la Figura 6-1.

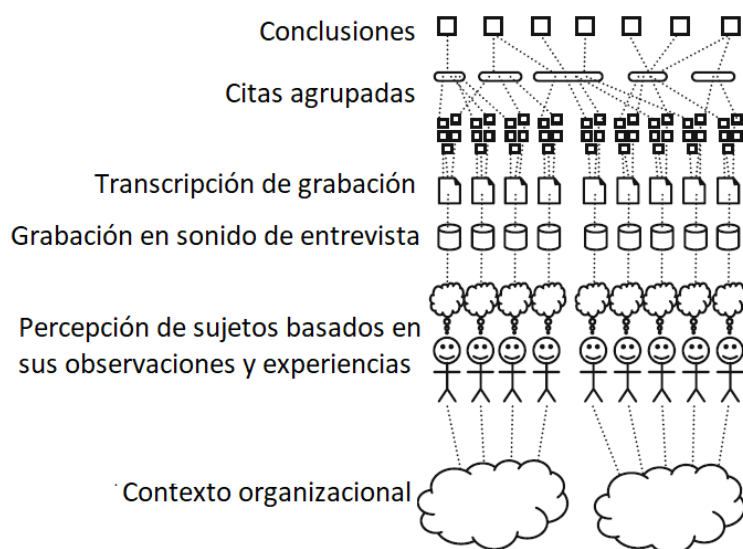


Figura 6-1: Abstracción de datos (Runeson & Höst, 2009).

En la etapa de análisis de este caso de estudio se han identificado las siguientes categorías:

- Labores de cobranzas
- Registro de datos de la empresa
- Subida de comprobantes para notificación

- Confirmación de pagos

Estas categorías permitirán identificar los resultados en las entrevistas, sobre cada individuo y conformar las conclusiones finales. El análisis ha sido realizado por dos investigadores, analizando tres entrevistas, sin herramientas especializadas.

El análisis de los datos recolectados ha permitido sintetizar aspectos relacionados a las categorías definidas, conforme se muestra en la Tabla 6-A.

Categoría	Síntesis de resultados de entrevistas
Labores de cobranzas	Disminución notable de carga laboral en el proceso de notificaciones a los clientes, al evitar buscar y contactar a los clientes; Eficiencia operativa por reducción drástica de tiempo requerido para notificación a los clientes sobre pagos vencidos o plazos máximos de pago alcanzados; Reducción en gastos operativos de notificación por vía telefónica u otros medios (SMS, visita personal); Confiabilidad generada por la automatización del proceso de notificación, al evitar posibles errores del personal.
Registro de datos de la empresa	Operación unificada de registro y actualización muy útil; Validación completa de información de empresa; Plantillas cuentan con detalle suficiente para correo electrónico y mensajes de texto.
Subida de comprobantes para notificación	Eficiencia al permitir subir comprobantes individualmente o en lote; Validación completa de comprobantes y datos relacionados a pagos de cada comprobante; Útil actualización de datos de contacto de cliente constante; Incertidumbre en la organización al no existir una confirmación de recepción de las notificaciones enviadas a los clientes.
Confirmación de pagos	Método eficiente al permitir confirmar múltiples pagos en una sola petición a la plataforma; Validación completa de datos relacionados a los pagos por confirmar de cada comprobante; Flexibilidad para utilizar números de identificación de pago propios de la empresa; Mayor cercanía de los clientes con la organización, al evidenciar menos suspensiones de servicio por incumplimiento de pagos.

Tabla 6-A: Síntesis de resultados (Entrevistas).

Por otro lado, las observaciones realizadas en la organización han permitido conocer la forma en la que se han venido manejando los procesos de cobranzas. En GSoft – Soluciones Informáticas el primer día de cada mes se emiten las facturas correspondientes a la utilización de los servicios que la organización brinda. Estas facturas son enviadas al correo electrónico de los clientes directamente desde el sistema de facturación electrónica que tiene la organización. Inmediatamente luego de realizar la emisión de estas facturas, se envía manualmente un correo electrónico y un mensaje de texto a cada uno de los clientes, indicándoles que se les ha enviado la factura correspondiente y se incluye información sobre cómo realizar el pago y el plazo máximo que dispone el cliente para cancelar dicho pago.

El plazo máximo del que disponen los clientes para cancelar los valores pendientes es de 10 días. Luego que ha transcurrido este tiempo, se verifican los pagos recibidos y por cada cliente que mantenga valores de pago pendientes se realiza una llamada telefónica indicándole que los servicios serán suspendidos por falta de pago. La organización ha entregado a los investigadores un documento de Excel en el cual se detallan las fechas en las cuales se ha realizado la cancelación de los valores pendientes, por cada factura emitida. Un fragmento de esta información se muestra en la Tabla 6-B.

Factura	Fecha de emisión	Fecha de pago	Días transcurridos	Servicio suspendido
001-100-000000019	08/08/2017	20/08/2017	12	SI
001-100-000000020	08/08/2017	18/08/2017	10	SI
001-100-000000021	01/09/2017	09/09/2017	8	NO
001-100-000000022	01/09/2017	09/09/2017	8	NO
001-100-000000023	02/10/2017	13/10/2017	11	SI
001-100-000000024	02/10/2017	11/10/2017	9	NO
001-100-000000025	01/11/2017	10/11/2017	9	NO
001-100-000000026	01/11/2017	11/11/2017	10	SI
001-100-000000028	01/12/2017	09/12/2017	8	NO
001-100-000000029	01/12/2017	12/12/2017	11	SI

Tabla 6-B: Información de pagos antes de la plataforma.

De manera similar, luego de haber realizado la integración de la organización con los servicios de notificación provistos por la plataforma, se ha solicitado a la organización la información que contiene los detalles sobre las fechas en las cuales se han realizado la cancelación de los valores pendientes, por cada factura emitida que se ha subido a la plataforma para despacho de notificaciones. En la Tabla 6-C se muestran los registros entregados a los investigadores:

Factura	Fecha de emisión	Fecha de pago	Días transcurridos	Servicio suspendido
001-100-000000030	01/01/2018	05/01/2018	4	NO
001-100-000000032	01/02/2018	11/02/2018	10	SI
001-100-000000033	01/02/2018	06/02/2018	5	NO
001-100-000000034	01/03/2018	06/03/2018	5	NO
001-100-000000035	01/03/2018	04/03/2018	3	NO

Tabla 6-C: Información de pagos después de la plataforma.

Se ha podido observar que desde la adopción de los servicios de notificación utilizando la plataforma, el tiempo transcurrido para la realización de los pagos se ha visto disminuido en un promedio de 10 días a un promedio de 5.4 días, en tan solo dos meses de aplicación del proceso de notificaciones automático. En la Figura 6-2 se muestra la tendencia que tienen los días transcurridos a partir del mes de enero, evidenciando una mejora en la eficiencia de las labores de cobranzas.

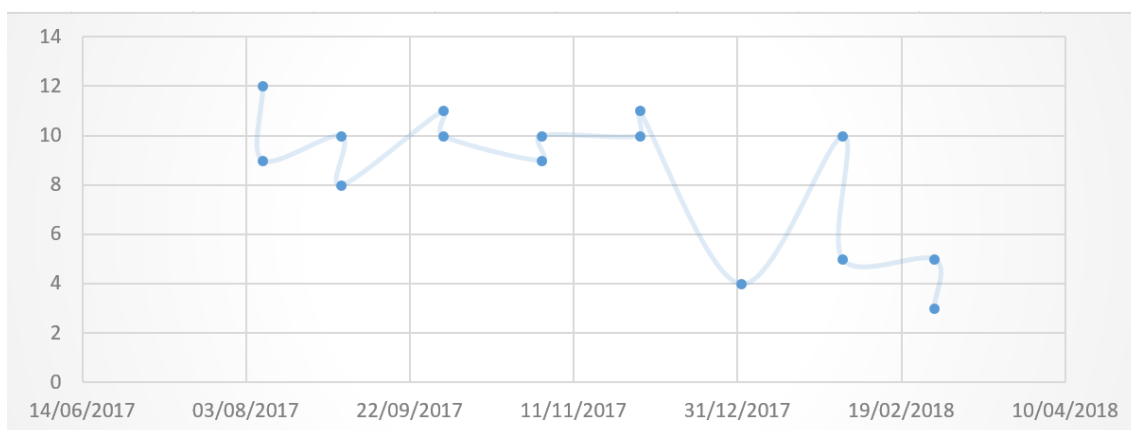


Figura 6-2: Tendencia de días transcurridos en pagos.

Por otro lado, antes de la adopción de los servicios de notificación automática con la plataforma, en el número de pagos que han sido realizados fuera del plazo definido (10 días) se puede evidenciar que en la distribución porcentual de la Figura 6-3 se han presentado un 70% de pagos tras suspensión de los servicios y un 30% de pagos dentro del plazo definido.

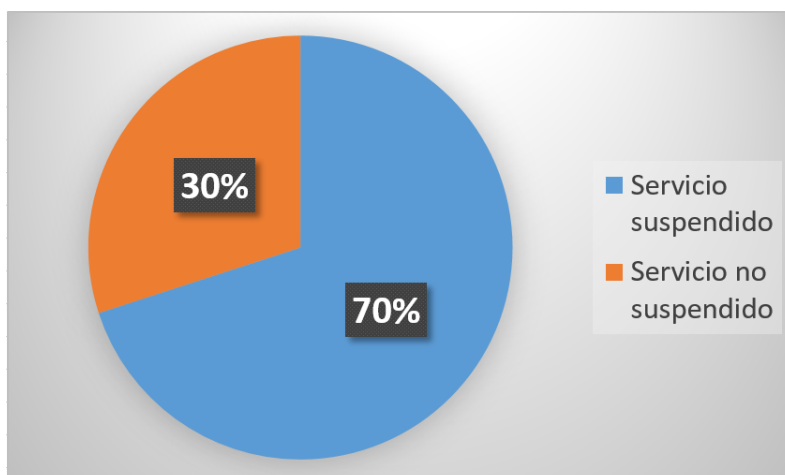


Figura 6-3: Suspensión de servicios, antes de la plataforma.

Sin embargo, una vez iniciado el proceso de notificaciones automáticas, los datos indican que solamente un 20% de pagos fueron realizados tras suspensión de servicios, mientras que el 80% de los pagos se han realizado previo a una suspensión de servicios, como se muestra en la Figura 6-4. Esto evidencia la capacidad de proveer servicios ininterrumpidos al mantener cercanía con los clientes, proporcionándoles información oportuna sobre los pagos que mantienen con la organización, evitando una eventual suspensión de los servicios.

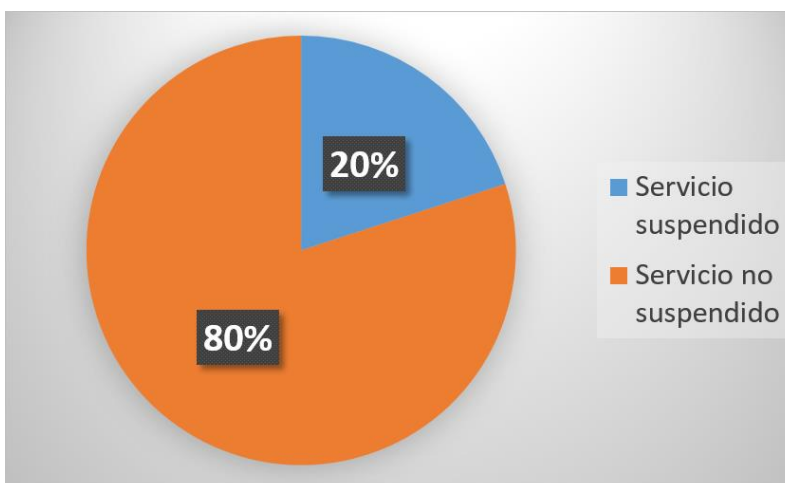


Figura 6-4: Suspensión de servicio después de la plataforma.

Por otro lado, se han realizado observaciones en la organización con el propósito de estimar tiempos necesarios para el envío de notificaciones, de forma manual y contrastarlos con los tiempos utilizados en notificaciones automáticas con la utilización de la plataforma en la nube. En este sentido, se ha considerado una sesión laboral normal consistente en el envío de notificaciones a 20 clientes sobre pagos vencidos. Antes del uso de la plataforma, la sesión consistía en redactar correos electrónicos y mensajes de texto para cada uno de



los clientes, con la información respectiva a cada una de las notificaciones (datos del cliente, valores pendientes, plazos máximos de pago, descripción de servicios brindados e información de puntos de pago). Tras la automatización del proceso de notificaciones con el uso de la plataforma, las labores se redujeron a la selección de facturas o pagos pendientes para notificación, confirmación de datos de contacto y envío hacia la plataforma.

En la Tabla 6-D se muestra la estimación de tiempos en los casos mencionados anteriormente. Esto evidencia una disminución drástica de los tiempos requeridos para llevar a cabo el proceso de notificaciones a los clientes. Con la utilización de la plataforma se requiere únicamente el 30% del tiempo requerido en notificaciones manuales de correo electrónico, y únicamente el 21% del tiempo requerido para notificaciones por SMS.

Operación	Tiempo estimado notificación manual	Tiempo estimado notificación plataforma
Envío de notificación de correo electrónicos a 20 clientes	50 minutos	15 minutos
Envío de notificación por SMS a 20 clientes	70 minutos	

Tabla 6-D: Tiempos estimados para notificación.

6.4 Discusión y conclusiones del caso de estudio

El presente caso de estudio ha permitido investigar los efectos producidos tras la utilización de servicios de notificación durante dos meses, mediante la plataforma desarrollada con el método propuesto en este trabajo de titulación. Específicamente, ha permitido descubrir debilidades del servicio brindado por la plataforma, así como aspectos positivos evidenciados en las labores del personal, analizando la disminución de tiempo transcurrido desde las notificaciones hasta la cancelación de los pagos, así como el ahorro de tiempo en el despacho de notificaciones.

El análisis realizado ha mostrado que los servicios provistos por la plataforma no presentan falencias en cuanto a su funcionamiento; por otro lado, su adopción ha generado efectos positivos en la organización mostrando un incremento de agilidad en la gestión de notificaciones a los clientes sobre sus pagos vencidos, reducción de esfuerzo y carga laboral por parte del personal de cobranzas de la organización, reducción de gastos relacionados a notificaciones manuales y generación de cercanía con los clientes al brindar información oportuna sobre sus pagos pendientes, reducción de errores, disminución de tiempos entre notificación y cancelación de los pagos. Estos beneficios proporcionan agilidad para la organización, lo cual afirma la hipótesis definida en el Capítulo 1, validando el método propuesto en este trabajo de titulación.



Capítulo 7

Conclusiones y trabajos futuros

El presente trabajo de titulación aporta con un método para desarrollar sistemas de información de pagos a organizaciones nacionales, lo que permite gestionar facturas electrónicas y enviar información a clientes. Para la consecución de este propósito se han planteado actividades inmersas en un proceso que sigue las bases de una metodología científica.

En este sentido, se ha presentado en primer lugar, un acercamiento hacia la base teórica de conceptos, tecnologías y elementos que se utilizan o guardan relación a este trabajo. Luego, se ha presentado un análisis del estado del arte en los temas relacionados a plataformas, facturación electrónica, notificaciones, computación en la nube, entre otras. Se ha realizado una revisión sistemática basándose en una metodología formal, lo cual ha permitido esclarecer la literatura existente en estos temas, seleccionando entre un total de 1202 artículos científicos solamente 39 artículos, que han permitido conformar un panorama claro sobre la falta de estudios que aborden el uso de notificaciones en sistemas de información de pagos a organizaciones, como el propuesto en este trabajo.

A continuación, se ha propuesto el método para el desarrollo de sistemas de información de pagos a organizaciones nacionales, el cual permite desarrollar una plataforma en la nube, que provee servicios de procesamiento y despacho de notificaciones, con lo cual las organizaciones pueden registrar sus datos, subir sus facturas, confirmar pagos y enviar información de pagos hacia sus clientes por medios digitales. Adicionalmente, se ha realizado una instanciación de una plataforma en la nube conforme el método propuesto, documentando a detalle los procesos de desarrollo y los resultados obtenidos durante cada fase del método.

Para el desarrollo de éste método se han utilizado las metodologías de desarrollo *ASD* y *MDA* que conforman la parte medular del método, el cual se ha desarrollado recuperando las mejores características de dichas metodologías y fusionándolas para la obtención de un método robusto. Para la selección de estas metodologías se ha realizado un análisis exhaustivo de múltiples metodologías de desarrollo ágil y desarrollo orientado a modelos, como se muestra en el capítulo 2, con lo cual se encontró que *ASD* y la metodología de Chitforoush presentan gran compatibilidad y ventajas que guardan gran relación con el propósito planteado en este trabajo.

El análisis realizado en el capítulo 3 ha mostrado la tendencia que existe en el desarrollo, donde la computación en la nube está fuertemente considerada por las capacidades que conlleva. Es por esto, que se ha visto pertinente la utilización de una infraestructura en la nube orientada a servicios, lo cual permite a las organizaciones integrar sus sistemas y extender sus capacidades con mayor agilidad y eficiencia, junto con una reducción de costos de equipamiento y mantenimiento. La computación en la nube requiere que las comunicaciones se lleven a cabo por un medio hostil, como es internet. Por este motivo, ha sido necesario tener en consideración los aspectos de seguridad como confidencialidad, autenticidad, integridad y no repudio de la información.



Finalmente, la aplicación de un caso de estudio en un contexto real, ha permitido validar que el método propuesto en este trabajo de titulación permite el desarrollo de un sistema de información de pagos y cumple con las funcionalidades requeridas y planteadas, proporcionando a una organización nacional agilidad en la gestión de cobranzas para recuperación de cartera. En este sentido, el caso de estudio ha mostrado que la utilización de la plataforma desarrollada con el método propuesto en este trabajo, ha incrementado la agilidad en el despacho de las notificaciones de pagos vencidos a los clientes y en la cancelación de dichos pagos. Así mismo, se ha mostrado una reducción de gastos relacionados a notificaciones manuales, y se ha evidenciado la creación de cercanía con los clientes por medio del envío de información oportuna sobre los pagos que mantienen con las organizaciones. Finalmente, el análisis ha mostrado que el uso de los servicios de notificación de la plataforma ha reducido la cantidad de errores que se presentaban antes, de forma manual.

De esta manera se han cumplido los objetivos planteados para este trabajo de titulación, logrando definir los requerimientos de las organizaciones de manera clara delimitando el alcance. Se ha analizado información existente sobre trabajos relacionados y tecnologías involucradas con el desarrollo del mismo. Esto ha permitido definir el método propuesto, detallando las actividades que involucra cada una de las fases. Se ha realizado una instanciación de la solución, la cual ha sido validada mediante la aplicación de un caso de estudio en una organización local.

En el desarrollo de este trabajo de titulación se han encontrado dificultades como la curva de aprendizaje elevada que representa el desarrollo con *MDA* y por ende la obtención del máximo de los beneficios asociados a las capacidades que esta forma de desarrollo puede proporcionar.

De igual forma, se han encontrado aspectos que no se han tratado a detalle por no encontrarse dentro del alcance y ámbito de éste trabajo de titulación. Este es el caso de la gestión de riesgos del proyecto y la recolección de requerimientos, de la primera fase del método propuesto. La ampliación de estos temas pueden ser considerados en trabajos adicionales, conforme el nivel de detalle que se requiera.

Se ha considerado propicio plantear como trabajo futuro la incorporación de funcionalidades de pasarela de pago, de manera que los clientes puedan efectuar los pagos directamente desde las notificaciones que se han recibido. De forma similar, se plantea como trabajo futuro la inclusión de otros medios de notificación como las llamadas robotizadas. Por otro lado, resulta conveniente realizar un análisis más profundo sobre los beneficios generados en la gestión de cobranzas de las organizaciones, en el valor agregado que proporciona, a manera de experimento con un número grande de organizaciones, lo cual daría más formalismo a la validación del método propuesto en este trabajo de titulación. Esto puede dar paso a líneas de investigación que involucren un análisis más estricto mediante la utilización de indicadores de rendimiento clave (*KPI* – Key Performance Indicator) en las organizaciones.



Referencias

- Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile software development methods: Review and analysis. *Espoo, Finland: Technical Research Centre of Finland, VTT Publications*, 112. <https://doi.org/10.1076/csed.12.3.167.8613>
- Abrahamsson, P., Warsta, J., Siponen, M. T., & Ronkainen, J. (2003). New Directions on Agile Methods: A Comparative Analysis. In *Software Engineering, 2003. Proceedings. 25th International Conference on* (Vol. 6, pp. 244–254). <https://doi.org/10.1109/ICSE.2003.1201204>
- Alexander, I. F., & Stevens, R. (2002). *Writing better requirements. Chemistry biodiversity* (Vol. 1). Retrieved from <http://onlinelibrary.wiley.com/doi/10.1002/cbdv.200490137/abstract%5Cnhttp://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Writing+Better+Requirements#0>
- Ali, N. (2016). *Adapting e-invoicing: benefits, challenges and future strategies to overcome challenges*. Vaasa, Finland. Retrieved from <https://www.theseus.fi/handle/10024/108392>
- Alippi, C., Pessina, F., & Roveri, M. (2005). An adaptive system for automatic invoice-documents classification. In *Proceedings - International Conference on Image Processing, ICIP* (Vol. 2, pp. 526–529). <https://doi.org/10.1109/ICIP.2005.1530108>
- Amazon. (2016). Amazon Web Service. Retrieved from <https://aws.amazon.com/>
- Amazon. (2018). Certificados X.509 - AWS IoT. Retrieved from https://docs.aws.amazon.com/es_es/iot/latest/developerguide/x509-certs.html
- Amazon Web Services. (2018). Amazon Elastic Compute Cloud Guía del usuario de instancias de Linux. Retrieved from https://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/ec2-ug.pdf#t2-instances
- Andreeff, A., Binmoeller, L. C., Boboch, E. M., Cerda, O., Chakravorti, S., Ciesielski, T., & Green, E. J. (2001). *Electronic bill presentment and payment - Is it just a click away? Economic Perspectives* (Vol. 25). Chicago, USA.
- Armbrust, M., Stoica, I., Zaharia, M., Fox, A., Griffith, R., Joseph, A. D., ... Rabkin, A. (2010). A view of cloud computing. *Communications of the ACM*, 53(4), 50. <https://doi.org/10.1145/1721654.1721672>
- Assimakopoulos, N. A., Riggas, A. N., & Kotsimpos, G. K. (2015). *A Systemic Approach for an Open Internet Billing System*. Piraeus, Grecia. Retrieved from <http://www.afscet.asso.fr/resSystemica/Crete02/Assimakopoulos, Riggas, Kotsibos.pdf>
- Astudillo, E., & Zambrano, J. (2006). *Sistema De Notificaciones Y Envío De Mensajes Sms Para La Comunidad De La Escuela Politécnica Del Ejército*. ESCUELA POLITÉCNICA DEL EJÉRCITO.
- Bauer, C., & King, G. (2005). *Hibernate in Action*. Manning (Vol. 4). Retrieved from <http://books.google.com/books?id=WCmSQgAACAAJ&pgis=1>
- Berlik, S. (2007). Eclipse Modeling Framework. <https://doi.org/10.1108/02641610810878585>
- Bézivin, J., Hammoudi, S., Lopes, D., & Jouault, F. (2004). Applying MDA approach for web service platform. *Proceedings of the 8th IEEE Intl Enterprise Distributed Object Computing Conf*



- (EDOC 2004), (1), 58–70. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1342505
- Boehm, B. W., Defense, T. R. W., Group, S., Boehm, H. W., Defense, T. R. W., & Group, S. (1987). A Spiral Model of Software Development and Enhancement. *Computer*, 21(May), 61–72. <https://doi.org/10.1109/2.59>
- Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., & Orchard, D. (2004). Web Services Architecture. *W3C Note*, 22(February), 1–98. <https://doi.org/10.1023/B:BTTJ.0000015492.03732.a6>
- Brun, M. H., & Lanng, C. (2006). Reducing barriers for e-business in SME's through an open service oriented infrastructure. In *Electronic commerce The new e-commerce: innovations for conquering current barriers, obstacles and limitations to conducting successful business on the internet - ICEC '06* (p. 403). <https://doi.org/10.1145/1151454.1151519>
- Cáceres, P., & Marcos, E. (2002). Marco metodológico para el desarrollo de sistemas de información Web basado en MDA. *Onekin.Org*, (April), 1–4. Retrieved from <http://www.onekin.org/webe03/Papers/2Caceres.pdf>
- Cáceres, P., Marcos, E., Vela, B., & Rey, J. (2008). A MDA-Based Approach for Web Information System Development, (April). Retrieved from https://www.researchgate.net/profile/Espananza_Marcos/publication/228847177_A_MDA-based_approach_for_web_information_system_development/links/5706232608aef745f71777d8.pdf
- Cárdenas, J. D., & García, A. (2018). Architecture for Electronic Notifications Services on the Cloud.
- Carpio, J., & Faicán, R. (2009). *Desarrollo de un sistema web de administración y visualización de alertas en tiempo real con notificación vía mensaje de texto y una aplicación móvil con geolocalización de emergencias médicas para la Cruz Roja – Loja*. Universidad Nacional de Loja.
- Cavanaugh, E. (2006). Web services: Benefits, challenges, and a unique, visual development solution. *Product Marketing Manager, Altova® WhitePaper*. Retrieved from http://www.altova.com/documents/whitepaper_webservices_2006.pdf
- Chester, T. M. (2001). Cross-platform integration with XML and SOAP. *IT Professional*, 3(5), 26–34. <https://doi.org/10.1109/6294.952977>
- Chinosi, M., & Trombetta, A. (2012). BPMN: An introduction to the standard. *Computer Standards and Interfaces*, 34(1), 124–134. <https://doi.org/10.1016/j.csi.2011.06.002>
- Chitforoush, F., Yazdandoost, M., & Ramsin, R. (2007). Methodology Support for the Model Driven Architecture. *14th Asia-Pacific Software Engineering Conference (APSEC'07)*, 454–461. <https://doi.org/10.1109/ASPEC.2007.58>
- Chiu, C. H., Wu, R. S., Tut, C. I., Lin, H. T., & Yuan, S. M. (2007). *Next generation notification system: Integrating instant messengers and Web service. 2007 International Conference on Convergence Information Technology, ICCIT 2007. Taiwan*. <https://doi.org/10.1109/ICCIT.2007.4420509>
- Chiu, J., & Lai, A. (2007). Modelling Payments Systems: A Review of the Literature. *Bank of Canada, Working Papers, 2007, 59 Pp., 59*. Retrieved from <http://ezproxy.library.tufts.edu/login?url=http://search.proquest.com/docview/5660449>



7?accountid=14434

- Chu, H., Chai, Y., Liu, Y., & Sun, H. (2014). *A novel E-Invoice Framework towards data-oriented taxation system. Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2014*. Beijing, China. <https://doi.org/10.1109/CSCWD.2014.6846849>
- Clear, C., & Cockburn, A. (2014). Crystal clear. *Nature*, 505(7485), 586. <https://doi.org/10.1111/j.1600-0560.2011.01732.x>
- Computerhope. (2018). E-mail. Retrieved from <https://www.computerhope.com/jargon/e/email.htm>
- Condes, L. (2005). Short Message Service).
- Cuadrado, J. S., Cánovas Izquierdo, J. L., & Molina, J. G. (2014). Applying model-driven engineering in small software enterprises. *Science of Computer Programming*, 89(PART B), 176–198. <https://doi.org/10.1016/j.scico.2013.04.007>
- Cuylen, A., Kosch, L., & Breitner, M. H. (2016). Development of a maturity model for electronic invoice processes. *Electronic Markets*, 26(2), 115–127. <https://doi.org/10.1007/s12525-015-0206-x>
- Delgado, J., & Marín, F. (2000). Evolución en los sistemas de gestión empresarial. *Economía Industrial*, 331.
- Dennehy, D., & Sammon, D. (2015). Trends in mobile payments research: A literature review. *Journal of Innovation Management Dennehy*, 3(1), 49–61.
- Diniz, E. H., Porto de Albuquerque, J., & Cernev, A. K. (2011). Mobile Money and Payment: A Literature Review Based on Academic and Practitioner - Oriented Publications (2001 - 2011). *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.2924669>
- Drossos, D. A., Giaglis, G. M., Vlachos, P. A., Zamani, E. D., & Lekakos, G. (2013). Consumer Responses to SMS Advertising: Antecedents and Consequences. *International Journal of Electronic Commerce*, 18(1), 105–136. <https://doi.org/10.2753/JEC1086-4415180104>
- E-administracion. (2018). Autoridad de Certificacion. Retrieved from <http://e-administracion.cea.es/autoridades>
- Evans, D., & Yen, D. C. (2006). E-Government: Evolving relationship of citizens and government, domestic, and international development. *Government Information Quarterly*, 23(2), 207–235. <https://doi.org/10.1016/j.giq.2005.11.004>
- Fenton, N. E. (1999). Software Metrics: Successes, Failures and New Directions. *The Journal of Systems and Software*, 47, 149–157. [https://doi.org/10.1016/S0164-1212\(99\)00035-7](https://doi.org/10.1016/S0164-1212(99)00035-7)
- Frankel, D., & Parodi, J. (2002). *White Paper Using Model-Driven Architecture™ to Develop*. Dublin, Ireland. Retrieved from <http://www.omg.org/attachments/pdf/WSMDA.pdf>
- García, A., & Cárdenas, J. D. (2018). A systematic literature review of electronic invoicing , platforms and notification systems. In *eDemocracy & eGovernment - ICEDEG 2018*.
- Ghazal, M., Ali, S., Al Halabi, M., Ali, N., & Al Khalil, Y. (2016). *Smart mobile-based emergency management and notification system. Proceedings - 2016 4th International Conference on Future Internet of Things and Cloud Workshops, W-FiCloud 2016*. Abu Dhabi, UAE. <https://doi.org/10.1109/W-FiCloud.2016.64>
- Gorschek, T., Garre, P., Larsson, S., & Wohlin, C. (2006). A model for technology transfer in



- practice. *IEEE Software*, 23(6), 88–95. <https://doi.org/10.1109/MS.2006.147>
- Highsmith, J. . (2000). Adaptive software development. Retrieved from http://ingenieriadesoftware.mex.tl/61154_ASD.html
- Highsmith, J. A., & Orr, K. (2001). Chapter 23. Adaptive Software Development. In *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems* (pp. 173–179).
- Horkoff, J., & Yu, E. (2006). I Estrella. Retrieved from <http://istar.rwth-aachen.de/tiki-index.php?page=iStarQuickGuide>
- Horkoff, J., & Yu, E. (2016). Interactive goal model analysis for early requirements engineering. *Requirements Engineering*, 21(1), 29–61. <https://doi.org/10.1007/s00766-014-0209-8>
- ISO/IEC. (1998). ISO/IEC 10746-1:1998. Retrieved from <https://www.iso.org/obp/ui/#iso:std:iso-iec:10746:-1:ed-1:v1:en>
- Ji, Z., Ganchev, I., Odroma, M., & Zhao, Q. (2014). A push-notification service for use in the UCWW. *Proceedings - 2014 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC 2014*. Limerick, Ireland. <https://doi.org/10.1109/CyberC.2014.88>
- Jimenez, T. M. (2008). *Guía de los Fundamentos para la Dirección de Proyectos (Guía del PMBOK). Management*. Retrieved from www.pmi.org
- Kalakota, R., & Robinson, M. (2001). *E-Business: roadmap for success*. Addison-Wesley information technology series.
- Kaliontzoglou, A., Boutsis, P., & Polemi, D. (2006). *eInvoice: Secure e-Invoicing based on web services*. *Electronic Commerce Research* (Vol. 6). Athens, Greece. <https://doi.org/10.1007/s10660-006-8678-6>
- Kent, S. (2002). Model Driven Engineering. https://doi.org/10.1007/3-540-47884-1_16
- Khasawneh, R. T., Rabayah, W. A., & Abu-shanab, E. A. (2013). E-Government Acceptance Factors : Trust and Risk. In *The 6th International Conference on Information Technology* (pp. 8–10). Retrieved from [http://sce.zuj.edu.jo/icit13/images/Camera Ready/E-Technology/613_final.pdf](http://sce.zuj.edu.jo/icit13/images/Camera%20Ready/E-Technology/613_final.pdf)
- Kiatruangkrai, P., Phusayangkul, P., Viniyakul, S., Prompoon, N., & Kanongchaiyos, P. (2010). *Design and Development of Real-Time Communication Content Management System for E-Commerce*. *2010 Second International Symposium on Data, Privacy, and E-Commerce*. Bangkok, Thailand. <https://doi.org/10.1109/ISDPE.2010.24>
- Kim, J., & Rohmer, S. (2012). *Electronic Billing vs. Paper Billing: Dematerialization, Energy Consumption and Environmental Impacts*. *Electronic Goes Green 2012+ (EGG)*. Troyes, France.
- Kitchenham, B., & Charters, S. (2007). *Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3*. *Engineering* (Vol. 45). Durham, UK. <https://doi.org/10.1145/1134285.1134500>
- Klensin, J. (2008). Simple Mail Transfer Protocol. <https://doi.org/10.17487/rfc5321>
- Kleppe, A., Warmer, J., & Bast, W. (2003). *[Anneke_Kleppe,_Jos_Warmer,_Wim_Bast]_MDA_Explained(BookZZ)*.
- Kontio, J. (1996). The Riskit Method for Software Risk Management, version 1.00 (CS-TR-3782;



UMIACS-TR-97-38). *Computer Science Technical Report*, (CS-TR-3705), 1–45.

- Kreuzer, S., Eckhardt, A., Bernius, S., & Krönung, J. (2013). *A unified view of electronic invoicing adoption: Developing a meta-model on the governmental level. Proceedings of the Annual Hawaii International Conference on System Sciences*. Frankfurt am Main, Germany. <https://doi.org/10.1109/HICSS.2013.67>
- Kwok, T., Laredo, J., & Maradugu, S. (2008). *A web services integration to manage invoice identification, metadata extraction, storage and retrieval in a multi-tenancy SaaS application. IEEE International Conference on e-Business Engineering, ICEBE'08 - Workshops: AiR'08, EM2I'08, SOAIC'08, SOKM'08, BIMA'08, DKEEE'08*. New York, USA. <https://doi.org/10.1109/ICEBE.2008.78>
- Lagzian, M., & Naderi, N. (2014). An Empirical Study of the Factors Affecting Customers ' Acceptance Intention of E-Invoice Services : The Case of Mashhad Electricity Distribution Company. In *Conf. on Electronic Governance and Open Society* (pp. 97–103). Mashhad, Iran. <https://doi.org/10.1145/2846012.2846050>
- Lee, K. J., Ju, J.-I., & Jeong, J. M. (2006). A payment & receipt business model in U-commerce environment. *Proceedings of the 8th International Conference on Electronic Commerce The New E-Commerce: Innovations for Conquering Current Barriers, Obstacles and Limitations to Conducting Successful Business on the Internet - ICEC '06*, 319. <https://doi.org/10.1145/1151454.1151507>
- Legner, C., & Wende, K. (2006). *Electronic bill Presentment and Payment*. Ecis. St. Gallen, Switzerland. Retrieved from <http://en.scientificcommons.org/58525657>
- Letelier, P., Canós, M., Sánchez, E., & Penadés, M. (2003). *Métodologías Ágiles en el Desarrollo de Software*. Valencia, España. Retrieved from http://www.carlosfau.com.ar/nqi/nqifiles/XP_Agil.pdf
- Lian, J. W. (2015). Critical factors for cloud based e-invoice service adoption in Taiwan: An empirical study. *International Journal of Information Management*, 35(1), 98–109. <https://doi.org/10.1016/j.ijinfomgt.2014.10.005>
- Liang, Y. (2006). *User-Website Interactive Communication Analysis in Web-Based Information System Development*. Paisley, U.K. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=B18CB140A20578F7BA37FA76446DED25?doi=10.1.1.93.635&rep=rep1&type=pdf>
- Macario, H. F., & Srirama, S. (2013). *Mobile cloud messaging supported by xmpp primitives. Proceeding of the Fourth ACM Workshop on Mobile Cloud Computing and Services*. Tartu, Estonia. <https://doi.org/10.1145/2482981.2482983>
- Martínez, S., Lamoth, L., Moreno, R., & Jacho, N. (2015). Análisis de la Transformación de Modelo CIM a PIM en el Marco de Desarrollo de la Arquitectura Dirigida por Modelos (MDA). *Revista Politécnica*, 36(3).
- Matson, M., & Ulieru, M. (2006). *Persistent Information Security: Beyond the e-Commerce Threat Model. Proceedings of the 8th International Conference on Electronic Commerce: The New e-Commerce: Innovations for Conquering Current Barriers, Obstacles and Limitations to Conducting Successful Business on the Internet*. Calgary, Canada. <https://doi.org/10.1145/1151454.1151500>
- Matus, A., Guerra, E., Fuertes, W., Gómez, M., Aules, H., Villacís, C., & Toulkeridis, T. (2017). *On the development of an electronic invoicing solution to integrate SMEs with a tax-collection*



- e-government-platform. 2017 4th International Conference on eDemocracy and eGovernment, ICEDEG 2017. Sangolquí, Ecuador. <https://doi.org/10.1109/ICEDEG.2017.7962518>
- Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology. *Nist Special Publication, 145*, 7. <https://doi.org/10.1136/emj.2010.096966>
- Menéndez Domínguez, V. H., & Castellanos Bolaños, M. E. (2008). Software Process Engineering Metamodel (SPEM). *Revista Latinoamericana de Ingenieria de Software, 3*(2), 92–100. <https://doi.org/10.18294/relais.2015.92-100>
- Mozilla. (2017). POST. Retrieved from <https://developer.mozilla.org/es/docs/Web/HTTP/Methods/POST>
- Muller, P. A., Studer, P., Fondement, F., & Bezivin, J. (2005). Platform independent Web application modeling and development with Netsilon. *Software and Systems Modeling, 4*(4), 424–442. <https://doi.org/10.1007/s10270-005-0091-4>
- Muñoz, L., Solarte, G., & Heredia, D. H. (2015). Una visión a la propuesta de Arquitectura Dirigida por Modelos * A vision to the proposal of Model Driven Architecture Uma visão a proposta de Model Driven Architecture. *UNIMAR, 33*(2), 270–279.
- Netter, M., & Pernul, G. (2009). *Integrating security patterns into the electronic invoicing process. Proceedings - International Workshop on Database and Expert Systems Applications, DEXA*. Regensburg, Germany. <https://doi.org/10.1109/DEXA.2009.12>
- Novillo, C. (2014). *Diseño E Implementación De Un Sistema De Seguridad Con Videocámaras, Monitoreo Y Envío De Mensajes De Alertas a Los Usuarios a Través De Una Aplicación Web Y/O Vía Celular Para Mejorar Los Procesos De Seguridad De La Carrera De Ingeniería En Sistemas Comp. Universidad de Guayaquil*. UNIVERSIDAD DE GUAYAQUIL. Retrieved from <http://repositorio.ug.edu.ec/bitstream/redug/6529/1/TesisCompleta-523.pdf>
- OMG. (2008). *Software & Systems Process Engineering Meta-Model Specification V2.0*. <https://doi.org/formal/2008-04-01>
- OMG. (2018). OMG - MDA. Retrieved from <http://www.omg.org/mda/>
- Oracle. (2005). Oracle GlassFish Server. Retrieved from <http://www.oracle.com/technetwork/es/middleware/glassfish/overview/index.html>
- Oracle. (2018). MySQL. Retrieved from <https://www.oracle.com/lad/mysql/index.html>
- Ordoñez, A. F., Suarez, P., & Villavicencio, M. (2016). *Metodologías ágiles en el desarrollo de software : un enfoque basado en Scrum y MDD*. Guayaquil, Ecuador.
- Paetsch, F., Eberlein, A., & Maurer, F. (2003). *Requirements engineering and agile software development. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003*. Calgary, Canada. <https://doi.org/10.1109/ENABL.2003.1231428>
- Penttinen, E., Hallikainen, P., & Salomäki, T. (2009). *Impacts of the implementation of electronic invoicing on buyer-seller relationships. Proceedings of the 42nd Annual Hawaii International Conference on System Sciences, HICSS*. Helsinki, Finland. <https://doi.org/10.1109/HICSS.2009.248>
- Penttinen, E., & Hyytiäinen, M. (2008). *The adoption of electronic invoicing in finnish private and public organizations. Ecis*. Helsinki, Finland.



- Peña Aguirre, D. R. (2016). *AUTOMATIZACIÓN PARA EL ENVÍO DE CORREOS ELECTRÓNICOS DE FORMA MASIVA Y PERSONALIZADA, INTEGRADO A LA PLATAFORMA ERP DE LA COMPAÑÍA CONSTRUCCIONES ISARCO*. Fundacion Universitaria los Libertadores. Retrieved from <http://repository.libertadores.edu.co/handle/11371/1017>
- Perez, J. D. (2000). Notaciones y lenguajes de procesos. Una visión global. *Estudios Filológicos*, (35), 109. <https://doi.org/10.4067/S0071-17132000003500023>
- Pintarić, N., Panjkota, A., & Zdrilić, I. (2010). *Effects of e-business between public and private sector based on the simulation of implementing e- Invoice*. *CECIIS Proceedings of the 21th Central European Conference on Information and Intelligent Systems*. Zadar, Croatia.
- Pons, C., Giandini, R., & Pérez, G. (2010). *Desarrollo de Software dirigido por modelos: Conceptos teóricos y su aplicación práctica*. Retrieved from http://sedici.unlp.edu.ar/handle/10915/26667%5Cnhttp://sedici.unlp.edu.ar/bitstream/handle/10915/26667/Documento_completo_.pdf?sequence=1
- Portal de Administración Electrónica. (2018). La firma electrónica. Retrieved from <http://firmaelectronica.gob.es/Home/Ciudadanos/Firma-Electronica.html>
- Potapenko, T. (2010). *Transition to e-invoicing and post-implementation benefits . Exploratory case studies*. Aalto University.
- Quintero, J. B., & Anaya, R. (2007). MDA y el papel de los modelos en el proceso de desarrollo de software. *Revista EIA*, ISSN 1794-1237, 131–146. Retrieved from https://campusvirtual.univalle.edu.co/moodle/pluginfile.php/141623/mod_resource/content/0/classnotes/DSI-NC01.pdf
- Radeski, D. (2011). *Electronic invoicing modeling*. Skopje, Macedonia. Retrieved from <http://ciit.finki.ukim.mk/data/papers/8CiIT/8CiIT-30.pdf>
- Raghupathi, W., & Umar, A. (2008). Exploring a model-driven architecture (MDA) approach to health care information systems development. *International Journal of Medical Informatics*, 77(5), 305–314. <https://doi.org/10.1016/j.ijmedinf.2007.04.009>
- Ramírez, L. del C., & Flórez, A. S. (2014). Desarrollo De Software Good Practice , a Solution for Better Software, 37–45.
- Riofrio Ramires, L. (2011). *DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO PROTOTIPO DE GESTOR DE NOTIFICACIONES A CLIENTES DE COMPAÑÍAS DE SEGUROS QUE INCLUYA AVISO DE VENCIMIENTOS DE PAGO, Y DE PÓLIZAS DE SEGUROS; POR MEDIO DE CORREO ELECTRÓNICO Y MENSAJES DE TEXTO A DISPOSITIVOS MÓVILES*, U. Tesis. Universidad de Guayaquil. Retrieved from <http://repositorio.ug.edu.ec/bitstream/redug/6793/1/Tesis Completa-310-2011.pdf>
- Rivera, D., & Pereira, J. (2011). *Tesis doctoral LOS CIBERMEDIOS EN ECUADOR: EVOLUCIÓN, ESTRUCTURA Y CIBERPARTICIPACIÓN*. Universidad de Santiago de Compostela. Retrieved from <https://minerva.usc.es/xmlui/bitstream/handle/10347/3712/RiveraRogel.pdf?sequence=1>
- Rodrigues Da Silva, A. (2015). Model-driven engineering: A survey supported by the unified conceptual model. *Computer Languages, Systems and Structures*, 43, 139–155. <https://doi.org/10.1016/j.cl.2015.06.001>
- Rofhök-björni, A. (2006). *Electronic Invoicing in Finland - attitudes towards electronic invoicing by financial managers in small- to mid-sized companies*. Swedish School of Economics and



Business Administration.

- Romero, J., Corrigan, J., & Mccauley, M. (2018). Applying a capabilities lens to manage the E & P portfolio. Retrieved from <https://www.strategyand.pwc.com/media/file/Strategic-portfolio-management.pdf>
- Roy, J., & Ramanujan, A. (2001). Understanding web services. *IT Professional*, 3(6), 69–73. <https://doi.org/10.1109/6294.977775>
- Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2), 131–164. <https://doi.org/10.1007/s10664-008-9102-8>
- Silvestrini, V. (1985). ¿Qué es la tecnología Java y para qué la necesito? Retrieved from https://www.java.com/es/download/faq/whatis_java.xml
- Smitha, K. K., Chitharanjan, K., & Thomas, T. (2012). Cloud based e-governance system A survey. *Procedia Engineering*, 38, 3816–3823. <https://doi.org/10.1016/j.proeng.2012.06.437>
- Spanic, D., Ristic, D., & Vrdoljak, B. (2011). *An electronic invoicing system. Proceedings of the 11th International Conference on Telecommunications*. Zagreb, Croatia.
- SRI. (2013). *Resolución NAC-DGERCGC13-00236*.
- SRI. (2017). *Resolucion No. NAC-DGERCGC17-00000430*.
- SRI. (2018a). Comprobantes Electrónicos Información básica. Retrieved from http://www.sri.gob.ec/documents/156146/0/pdf+FACTURACION+ELECTRONICA+V1_out_03_03_2015.pdf/489fb78d-5e8d-4a01-808f-b4417d1842dc
- SRI. (2018b). factura electronica. Retrieved March 1, 2018, from <http://www.sri.gob.ec/web/guest/que-es-el-sri>
- SRI. (2018c). Facturacion electronica. Retrieved March 1, 2018, from <http://www.sri.gob.ec/web/guest/facturacion-electronica#información>
- Stojanovic, Z., Dahanayake, A., & Sol, H. (2003). Modeling and Architectural Design in Agile Development Methodologies. *Evaluation of Modeling Methods in System Analysis and Design*, 1–10.
- Sundström, J. (2006). *Adoption of Electronic Invoicing in SMEs*. Lulea University of Technology. Retrieved from <https://pure.ltu.se/ws/files/30990780/LTU-EX-06082-SE.pdf>
- Sungkur, R. K., Gangabaksh, Y., & Rutah, N. (2016). *Cloud-based cross-platform push notification system for more informed learners. 2016 IEEE International Conference on Emerging Technologies and Innovative Business Practices for the Transformation of Societies, EmergiTech 2016*. Mauritius. <https://doi.org/10.1109/EmergiTech.2016.7737344>
- Suwisuthikasem, S., & Tangsripairoj, S. (2008). *E-Tax Invoice System using Web Services technology: A case study of the Revenue Department of Thailand. Proc. 9th ACIS Int. Conf. Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2008 and 2nd Int. Workshop on Advanced Internet Technology and Applications*. Bangkok, Thailand. <https://doi.org/10.1109/SNPD.2008.83>
- Tanna, G. B., Gupta, M., Rao, H. R., & Upadhyaya, S. (2005). Information assurance metric development framework for electronic bill presentment and payment systems using transaction and workflow analysis. *Decision Support Systems*, 41(1), 242–261. <https://doi.org/10.1016/j.dss.2004.06.013>













- Taylor, P., Chau, P. Y. K., Hui, K. L., Chau, P. Y. K., & Hui, K. L. (2009). Journal of Organizational Computing and Determinants of Small Business EDI Adoption : An Empirical Investigation Determinants of Small Business EDI Adoption : An Empirical Investigation. *Journal of Organizational Computing and Electronic Commerce*, 9392(February 2015), 37–41. <https://doi.org/10.1207/S15327744JOCE1104>
- Taylor, P., Leonard, L. N. K., & Cronan, T. P. (2009). A Study of the Value and Impact of Electronic Commerce : Electronic Versus Traditional Replenishment in Supply Chains A Study of the Value and Impact of Electronic Commerce : Electronic Versus Traditional Replenishment in Supply Chains, (August 2015), 37–41. <https://doi.org/10.1207/s15327744joce1204>
- Tseng, Y., Chang, C., & Tseng, J. C. R. (2007). Modeling and Implementation of Object-Oriented E-Commerce. In *E-Commerce Technology and the 4th IEEE International Conference on Enterprise Computing* (p. 8). Hsinchu, Taiwan. Retrieved from <http://ieeexplore.ieee.org/document/4285234/>
- Tung, L. L. (2004). *Service Quality and Perceived Value's Impact on Satisfaction, Intention and Usage of Short Message Service (SMS)*. *Information Systems Frontiers* (Vol. 6). Singapore. <https://doi.org/10.1023/B:ISFI.0000046377.32617.3d>
- tutorialspoint.com. (2016). *Adaptive Software Development*. Time. <https://doi.org/10.1017/CBO9781107415324.004>
- Vanjak, Z., Mornar, V., & Magdalenic, I. (2008). Deployment of e-invoice in croatia. In *Proceedings of the Third International Conference on Software and Data Technologies* (pp. 348–354). Zagreb, Croatia.
- Veselá, L., & Radiměřský, M. (2014). The Development of Electronic Document Exchange. *Procedia Economics and Finance*, 12(March), 743–751. [https://doi.org/10.1016/S2212-5671\(14\)00401-8](https://doi.org/10.1016/S2212-5671(14)00401-8)
- Voigt, B. J. . J. (2004). *Dynamic System Development Method. Technology*. Zurich, Sweden. Retrieved from https://files.ifi.uzh.ch/rerg/arvo/courses/seminar_ws03/14_Voigt_DSMD_Ausarbeitung.pdf
- Voutilainen, V., & Pento, T. (2003). Electronic Invoice Processing As a Tool. *FRONTIERS OF E-BUSINESS RESEARCH 2003*, 215–229.
- W3C. (1999). Hypertext Transfer Protocol. Retrieved from <https://www.w3.org/Protocols/rfc2616/rfc2616.html>
- W3C. (2007). SOAP Version 1.2. Retrieved from <https://www.w3.org/TR/soap12/#intro>
- Williams, R. C., Pandelios, G. J., & Behrens, S. G. (1999). *Software Risk Evaluation (SRE) Method Description (Version 2 . 0)*. Evaluation. Pittsburgh, Pensilvania.
- www.chakray.com. (2016). SOAP - ¿Qué diferencias hay entre REST y SOAP? Retrieved from <https://www.chakray.com/que-diferencias-hay-entre-rest-y-soap/>
- Yen, B. P.-C., & Ng, E. O. S. (2003). The Impact of Electronic Commerce on Procurement. *Journal of Organizational Computing & Electronic Commerce*, 13(3/4), 167–189. <https://doi.org/10.1080/10919392.2003.9681159>
- Yu, E. S. K. (1997). *Towards modelling and reasoning support for early-phase\requirements engineering*. *Proceedings of ISRE '97: 3rd IEEE International Symposium on Requirements Engineering*. Toronto, Canada. <https://doi.org/10.1109/ISRE.1997.566873>












Zissis, D., & Lekkas, D. (2011). Securing e-Government and e-Voting with an open cloud computing architecture. *Government Information Quarterly*, 28(2), 239–251. <https://doi.org/10.1016/j.giq.2010.05.010>

1. Anexo A: Símbolo, Términos y Descripción de SPEM 2.0

Símbolo	Termino original	Termino	Descripción
	Start	Inicio	Inicio del proceso
	End	Finalización	Finalización inmediata del proceso
	Work Product	Producto de trabajo	Es consumido, producido o modificado por una tarea.
	Work Product Use	Producto de trabajo en uso	Producto de trabajo dentro del contexto de una actividad determinada.
	Guideline	Guia/Directriz	Es un elemento de proceso, el cual provee información adicional relacionada con otros elementos.
	Role	Rol	Conjunto de habilidades, competencias y responsabilidades de un individuo o grupo.
	Role Use	Rol en uso	Un rol en el contexto de una actividad determinada.
	Task	Tarea	Describe una unidad de trabajo asignable y gestionable. Define el trabajo llevado a cabo por las instancias de roles.
	Task Use	Tarea en uso	Representa una tarea dentro de un proceso determinado.
	Group	Grupo	Para agrupar distintos elementos.




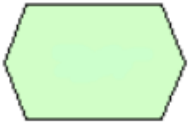
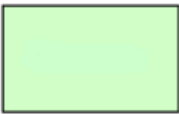
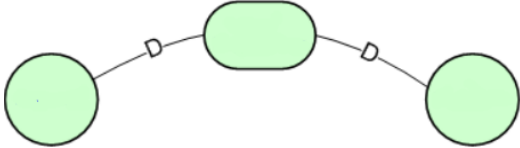
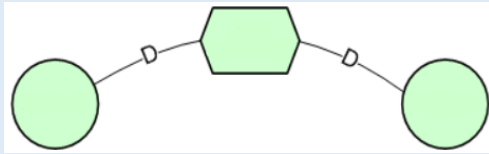
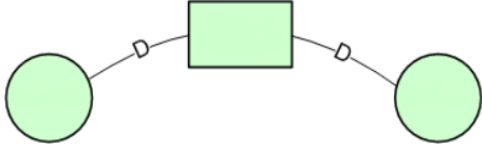
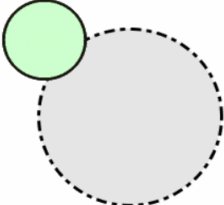
Anexo A: Nomenclatura SPEM 2.0 (Perez, 2000)..

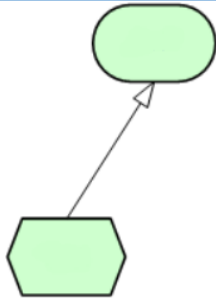
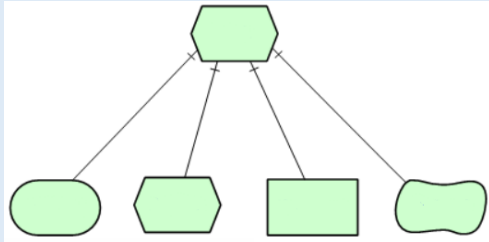


2. Anexo B: Símbolo, Tipo y Descripción de BPMN

Símbolo	Tipo	Descripción
	Evento de Inicio (Start Event)	Representa el inicio de un proceso
	Evento de Intermedio (Intermediate Event)	Detiene el flujo hasta que ocurra una condición o dispara acciones de excepción
	Evento de Fin (End Event)	Indica cuando finaliza un proceso en ejecución
	Flujo de secuencia (Sequence Flow)	Para indicar el orden en el cuál son ejecutadas las actividades del proceso de negocio.
	Pool	Para indicar los participantes en el proceso.
	Lane	Es una partición de Pool, permite clasificar las actividades.
	Grupo (Group)	Para agrupar distintos elementos del diagrama.
	Compuerta Exclusiva (Exclusive Gateway)	Divergente: son decisiones que toma el usuario del sistema para decir el camino a seguir.
	Tarea (Task)	Unidad de trabajo dentro de un proceso.

Anexo B: Nomenclatura BPMN (Perez, 2000).

3. Anexo C: Símbolo, Tipo y Descripción de i*

Simbología	Tipo	Descripción
	Actor	Entidades activas que realizan acciones para alcanzar metas.
	Objetivo	Es el deseo de un actor, representa una intención.
	Objetivo blando	Son similares a los objetivos, sin embargo el criterio para su cumplimiento no está del todo claro y puede considerarse un nivel no tan preciso de satisfacción.
	Tarea	Representa lo que el actor quiere conseguir de alguna manera. Puede ser descompuesta en sub-elementos.
	Recurso	Es la representación de la provisión de alguna entidad. Se obvian las dudas sobre el cómo, de dicha provisión.
	Dependencia de objetivo	Muestra la relación de dependencia entre dos actores para alcanzar un objetivo.
	Dependencia de tarea	Representa la relación de dependencia entre dos actores para llevar a cabo una determinada actividad o tarea.
	Dependencia de recurso	Se representa a un actor como proveedor de un determinado recurso hacia otro actor.
	Límites de actor	Todo lo que se encuentra dentro del círculo entrecortado se considera como deseos o intenciones específicas del actor.

	<p>Enlace Medio-Fin</p>	<p>Indican la relación entre un fin y el medio para obtenerlo. Se da entre una tarea y un objetivo</p>
	<p>Enlaces de descomposición</p>	<p>Una tarea puede ser descompuesta por medio de enlaces de descomposición en sub-objetivos, sub-tareas, recursos y objetivos blandos.</p>
	<p>Enlace de contribución positivo <i>Help</i></p>	<p>Simboliza una contribución parcial positiva, no suficiente por sí misma para alcanzar un objetivo.</p>
	<p>Enlace de contribución negativo <i>Hurt</i></p>	<p>Simboliza una contribución parcial negativa, no suficiente por sí misma para evitar un objetivo.</p>

Anexo C: Nomenclatura i* (Horkoff & Yu, 2006).



4. Anexo D: Oficio para caso de estudio en GSoft



Edgar García Sempértegui
Desarrollador de software
Cuenca, Ecuador
Tel. (móvil): 09883275338
gsoft.solucionesinformaticas@gmail.com
www.gsoft.ec

Cuenca, 29 de diciembre de 2017


Señor
Juan Diego Cárdenas
Universidad de Cuenca

De mi consideración:

Atendiendo a su petición para la realización de un caso de estudio en GSoft – Soluciones Informáticas, para la validación del trabajo de titulación “Método para desarrollar sistemas de información de pagos a organizaciones nacionales, permitiendo gestionar facturas electrónicas y enviar información a clientes”, nos complace informarle que hemos accedido a su petición. En este sentido, ponemos a su disposición la información que requieran, para lo cual se deberán coordinar previamente reuniones con el personal correspondiente. Es conveniente recalcar que dicha información será entregada para propósitos exclusivos de ésta solicitud y no puede ser publicada la información concerniente a nuestros clientes.

Esperando que ésta información le sea de utilidad para iniciar con sus gestiones en la empresa, quedamos a su disposición.

Atentamente


Edgar García Sempértegui,
GSoft
Soluciones Informáticas

Anexo D: Oficio de aceptación por parte de GSoft.



5. Anexo E: Entrevistas del caso de estudio

Entrevista 1

Entrevistado: Encargado de cobranzas

En esta sección se utiliza una escala del 1 al 5 para que el personal de cobranzas valore cada una de las preguntas, como se muestra a continuación:

1. Nada 2. Muy poco 3. Moderadamente 4. Bastante 5. Completamente

¿La utilización de los servicios de la plataforma para la gestión de notificaciones ha facilitado sus labores de cobranzas?

Yo diría que 4 (Bastante)

¿La utilización de los servicios de la plataforma para la gestión de notificaciones le ha permitido reducir el tiempo operacional de notificar a sus clientes?

Igualmente 4 (Bastante)

¿En qué medida usted considera que el uso de los servicios de la plataforma para la gestión de notificaciones ha disminuido la tasa de errores?

Un 3 (Moderadamente)

Esta sección de la entrevista permite al personal de cobranzas dar su opinión de forma abierta, permitiendo explorar y descubrir información importante sobre los aspectos a evaluar.

¿Cómo ha sido su experiencia en la gestión de cobranzas utilizando la plataforma para el envío de notificaciones?

He notado que se ha disminuido en gran medida la carga existente en el proceso de notificación a los clientes con el que se venía trabajando anteriormente.

¿Qué aspectos le han parecido interesantes o útiles en el proceso de notificaciones?

Me ha parecido bastante interesante y útil la posibilidad de enviar a la plataforma una o incluso múltiples facturas para notificación. Se nota la plataforma ha sido programada considerando una buena validación de los datos que enviamos.

¿Qué experiencia tuvo en la utilización de la operación de registro de organización?

Me gustó que se ha incluido la posibilidad de que definamos las plantillas o mensajes que queremos mostrar a nuestros clientes, conforme correspondan a notificaciones de correo y mensajes de texto.



¿Qué experiencia tuvo en la utilización de la operación de subida de comprobantes para notificación?

Los datos de los clientes se actualizan conforme se realiza cada envío, lo cual es bastante útil pues tenemos casos donde nuestros clientes cambian de número o de correo electrónico, incluso pueden estar mal copiados.

¿Qué experiencia tuvo en la utilización de la operación de confirmación de pagos?

Es bastante sencillo realizar esta acción, a diferencia de antes que debíamos coordinar con contabilidad y apuntes manuales. Ahora simplemente se accede al sistema y se confirma los pagos, con lo cual las notificaciones quedan desactivadas.



Entrevista 2

Entrevistado: Auxiliar de cobranzas

En esta sección se utiliza una escala del 1 al 5 para que el personal de cobranzas valore cada una de las preguntas, como se muestra a continuación:

1. Nada 2. Muy poco 3. Moderadamente 4. Bastante 5. Completamente

¿La utilización de los servicios de la plataforma para la gestión de notificaciones ha facilitado sus labores de cobranzas?

Si, 5 (completamente)

¿La utilización de los servicios de la plataforma para la gestión de notificaciones le ha permitido reducir el tiempo operacional de notificar a sus clientes?

Si, 4 (bastante)

¿En qué medida usted considera que el uso de los servicios de la plataforma para la gestión de notificaciones ha disminuido la tasa de errores?

Moderadamente, 3

Esta sección de la entrevista permite al personal de cobranzas dar su opinión de forma abierta, permitiendo explorar y descubrir información importante sobre los aspectos a evaluar.

¿Cómo ha sido su experiencia en la gestión de cobranzas utilizando la plataforma para el envío de notificaciones?

Me ha facilitado muchísimo el trabajo que venía realizando. Parte de mis labores consistían en la coordinación entre contabilidad y el encargado de cobranzas, por lo cual resultaba bastante tedioso y cansado llevar los registros de los clientes que debían ser notificados y de los que ya habían realizado sus pagos.

¿Qué aspectos le han parecido interesantes o útiles en el proceso de notificaciones?

Es un sistema bastante novedoso pues agiliza las labores que llevamos a cabo notificando a más clientes en menor tiempo y nos proporciona tranquilidad cuando cargamos los datos de nuestros clientes y de las deudas que mantienen con nuestra organización.

¿Qué experiencia tuvo en la utilización de la operación de registro de organización?

Esta operación de registro le corresponde al encargado de cobranzas, sin embargo, me parece muy útil que el sistema tenga la capacidad de que cualquier empresa ingrese sus datos o preferencias.



¿Qué experiencia tuvo en la utilización de la operación de subida de comprobantes para notificación?

He tenido una experiencia bastante satisfactoria. En mi caso la complejidad de mis tareas se han disminuido a únicamente coordinar y presentar al encargado de cobranzas el resumen contable de los clientes que mantienen pagos vencidos.

¿Qué experiencia tuvo en la utilización de la operación de confirmación de pagos?

Es bastante importante que el sistema nos permite manejar nuestros propios códigos de pagos vencidos, pues esto nos permite identificar con nuestros sistemas contables los pagos que deben ser confirmados.



Entrevista 3

Entrevistado: Contador

En esta sección se utiliza una escala del 1 al 5 para que el personal de cobranzas valore cada una de las preguntas, como se muestra a continuación:

1. Nada 2. Muy poco 3. Moderadamente 4. Bastante 5. Completamente

¿La utilización de los servicios de la plataforma para la gestión de notificaciones ha facilitado sus labores de cobranzas?

Si yo diría que 3 (moderadamente)

¿La utilización de los servicios de la plataforma para la gestión de notificaciones le ha permitido reducir el tiempo operacional de notificar a sus clientes?

Bastante (4)

¿En qué medida usted considera que el uso de los servicios de la plataforma para la gestión de notificaciones ha disminuido la tasa de errores?

5 (completamente)

Esta sección de la entrevista permite al personal de cobranzas dar su opinión de forma abierta, permitiendo explorar y descubrir información importante sobre los aspectos a evaluar.

¿Cómo ha sido su experiencia en la gestión de cobranzas utilizando la plataforma para el envío de notificaciones?

Se han producido cambios en la recuperación de cartera, especialmente en mis labores de contabilidad he visto que las cancelaciones de valores pendientes los clientes las realizan más pronto, incluso la suspensión de servicios desde que utilizamos el sistema de notificaciones es menor. En comparación a como nos manejábamos antes, ahora existe una mayor cantidad de clientes que realizan sus pagos a tiempo, y se contactan para cualquier consulta sobre sus pagos gracias a las notificaciones que les enviamos.

¿Qué aspectos le han parecido interesantes o útiles en el proceso de notificaciones?

Es útil para nuestra organización que podamos incluir en los correos y mensajes de texto, información sobre los puntos de pago donde los clientes pueden acercarse a cancelar.

¿Qué experiencia tuvo en la utilización de la operación de registro de organización?

El sistema está bien pensado pues considera en los datos que enviamos sobre nuestra organización la veracidad, lo cual es importante pues como organización debemos estar seguros que no van a existir conflictos con terceros que utilicen este sistema.



¿Qué experiencia tuvo en la utilización de la operación de subida de comprobantes para notificación?

Fuera bueno que se incluyera la posibilidad de realizar los pagos de forma electrónica (en línea). También se ha visto la necesidad que el sistema nos avise o nos confirme cuando se han enviado correctamente las notificaciones.

¿Qué experiencia tuvo en la utilización de la operación de confirmación de pagos?

Se ha facilitado la coordinación que realizamos con el auxiliar de cobranzas, puesto que al verificar en nuestras cuentas que los valores pendientes han sido cancelados, simplemente debemos indicarle los códigos de los pagos que se deben finalizar e inmediatamente esto culmina con el proceso.

6. Anexo F: Artículos de investigación 1

A systematic literature review of electronic invoicing, platforms and notification systems

Priscila Cedillo
Computer Science Department
University of Cuenca
Cuenca, Ecuador
priscila.cedillo@ucuenca.edu.ec

Andrés García
Systems Engineering School
University of Cuenca
Cuenca, Ecuador
andres.garcia@ucuenca.ec

Juan Diego Cárdenas
Systems Engineering School
University of Cuenca
Cuenca, Ecuador
jdiego.cardenas@ucuenca.ec

Alexandra Bermeo
Faculty of Engineering
University of Cuenca
Cuenca, Ecuador
alexandra.bermeo@ucuenca.edu.ec

Abstract— In the XXI century, electronic commerce is considered as an alternative to reduce costs, increase the productivity, among other benefits, all of them perceived through the digitalization of billing. Although there is a large amount of literature on this field, such literature is not focused on electronic invoicing platforms with notification delivery. Thus, the present systematic literature review has been performed, it is based on a reliable, auditable and rigorous methodology in order to know and clarify the state of the art in the subject associated to electronic invoicing, technological aspects related and notification systems. The analysis of the 39 selected papers from a total of 1202, shows the absence of studies that relate the use of notifications into e-invoicing systems.

Keywords— e-invoicing, electronic billing, notification, security, sms, mail, platform, review.

I. INTRODUCTION

The electronic invoicing concept emerges as a modern, reliable and efficient method for handling and processing invoices of products, services, taxes and other types of merchandise, without the need of paper [1]. Electronic invoicing is given as an information system service that collects information from a business transaction and transmits it over a network [2].

Nowadays, with the development of smartphones, people interact actively with those devices. Therefore, users seek to have all personal and work information synchronized and at their fingertips, turning the use of emails, instant messaging applications, and SMS as a regular way to be informed. These are key channels for the delivery of business information, from Business to Consumers (B2C).

On the other hand, extensible markup language, XML, is a format used widely to interchange information about invoices in a digital manner [3], [4], [5], [6], [7]. Through the use of XML, there are many implementations that use web services for receiving, storing, processing and sending information to tax collectors [4], [6]. Said invoices can be sent to the customer e-mail like in [4], which includes the invoice in XML and a version of the document in PDF format.

Also, the security represents an important non-functional characteristic to be considered in e-invoicing solutions, because they usually handle sensitive information of clients. Firstly, this characteristic should be based on rules and laws of each country in terms of electronic invoicing. Also, it is important to

take into account certain measures for a correct handling of information. These essential requirements allow authentication, content integrity, non-repudiation of origin and receiver, confidentiality and privacy, sequence integrity and corresponding policies [3], [7]. As it is shown in [8], the use of the electronic invoicing allows: (i) to save time and money, (ii) control over payments, (iii) the generation of universal payment mechanisms, (iv) privacy and security, (v) reliability and (vi) dispute resolution. Implementing electronic invoicing in a company or business has a significant impact on the exchange of information and operational links between B2C, creating a more lasting and committed relationship [9].

On the other hand, exploring the benefits of electronic invoicing versus traditional paper invoicing show the reduction of costs by the reduction of transportation costs and saving time by reducing workflow [10]. Even the environmental impacts of electronic invoicing are much lower than traditional invoicing, mainly due to energy consumption [11]. At an organizational or business level, electronic invoicing improves control of billing processes by increasing productivity in invoice handling units, reducing processing time, reducing paper consumption, among others [12].

In any case, as far as it is known, there have not been found secondary studies about e-invoicing systems with notifications. It is important to know how the existing solutions of e-invoicing systems, associated technologies, security considerations and notifications delivery mechanisms have been proposed. Therefore, in this paper we present a systematic review of literature, which addresses the state of the art in terms of e-invoicing systems along with the delivery of notifications by digital means. The results obtained with this research show that the relationship between e-invoicing systems and the use of digital notifications has not been addressed yet.

To carry out this research it has been used the methodology proposed by Kitchenham [13], which consists on three stages: planning, execution and documentation. This paper is structured as follows: firstly, section 2 addresses the works related to the research topic. Next, in Section 3, the systematic literature review is presented, including the protocol, execution and results; and finally, Section 4 presents the conclusions and further work.

7. Anexo G: Artículos de investigación 2

Architecture for Electronic Notifications Services on the Cloud

A Model Driven Approach

Juan Diego Cárdenas
Systems Engineering School
University of Cuenca
Cuenca, Ecuador
jdiego.cardenas@ucuenca.ec

Andrés García
Systems Engineering School
University of Cuenca
Cuenca, Ecuador
andres.garcia@ucuenca.ec

Priscila Cedillo
Computer Science Department
University of Cuenca
Cuenca, Ecuador
priscila.cedillo@ucuenca.edu.ec

Alexandra Bermeo
Faculty of Engineering
University of Cuenca
Cuenca, Ecuador
alexandra.bermeo@ucuenca.edu.ec

Abstract—Cloud computing technology is being adopted by organizations driven by their need to reduce operational costs, and to deliver scalable and flexible services. Solutions deployed on the cloud are a great point of interest in the commercial and governmental area for e-commerce transactions. Here, electronic invoicing applications play a key role in commercial activities, being a branch of vital importance for the business activities. In order to accomplish certain tasks (e.g., collection of overdue invoices, notifications about payment plans) technological solutions have been proposed. However, it is important to establish architectures that support developers in creating said solutions in order to send messages about issues of interest for organizations in a proper manner. Thus, this paper proposes a services architecture for the creation of this type of platforms, which allows the management of notifications with information provided by companies. Such service could be provided by the use of cloud computing technologies in order to allow ubiquitous and heterogeneous access in a flexible and easy to configure manner, in which it will be possible to change the notifications' requirements without a reimplementation of the application. Here, a model driven engineering can be a suitable solution to be taken into consideration.

Keywords—cloud computing; security; protocol; web service; e-invoicing; notification; SMS; mail.

I. INTRODUCTION

Nowadays e-business plays a critical role in the maintenance and management of business information along the supply chain [1]. It covers a broad field, from which several branches are derived; this includes the world of e-invoicing. The Electronic Data Interchange (EDI) was firstly implemented between companies, and later, it was used to transmit electronic invoices between individuals, companies and government offices. Therefore, it has become the backbone of electronic commerce. The need to transmit information has allowed that so many countries use cloud computing as a means to provide invoicing services to the population [1].

The nature of the electronic invoicing allows to send, receive and process invoices without manual intervention [2]. According to Muller [3], an electronic invoice is a modern, reliable, cost-effective and essentially paperless method of managing and processing invoices generated by a business, which has been implemented in many countries. It is an electronic voucher, like an invoice; moreover, it is a document that meets the legal and regulatory requirements that are required for all proof of sale. Therefore, it guarantees

the authenticity of its origin and the integrity of its content [4]. Also, it has the same accounting benefits as a traditional commercial invoice.

Overdue accounts and clients with arrears in payments are a headache for many companies and organizations. Therefore, one of the most important objectives of an organization is the timely collection of services rendered or products sold to customers. The recovery of past due portfolio is a process that requires an efficient planning, which allows the successful benefit of the liquidity and solvency of a company, business or organization. Thus if this concern is not a priority for companies, this could break or generate losses of money, time and effort [5]. However, there are few companies that have a system of electronic notifications to inform customers about their situation related to their debts and overdue bills. Those solutions are restricted to the specific requirements of the enterprises, therefore if a requirement changes, the e-notification infrastructure needs to be re-implemented.

On the other hand, the use of web services and cloud technologies allow companies to incorporate new technological solutions at lower cost, focusing their development on the core of their business. Also, they add value aspects; maintaining the investment of the ERP or the legacy systems. Another feature is to integrate business processes with partners at lower cost when sharing their business processes independently of their own technologies [6].

According to Booth [7] a web service is "a software system designed to support the interoperable interaction of machine-to-machine through a network. It has an interface described in a machine-processable format (WSDL specification). Other systems interact with the web service in the manner prescribed by its description using Simple Object Access Protocol (SOAP) messages, usually transmitted via HTTP with an XML serialization along with other standards related to the Web.

Similarly, cloud computing, as defined by The National Institute of Standards and Technology (NIST), is "a model to allow ubiquitous, convenient and on-demand network access to a shared pool of configurable computing resources that can be provisioned and released quickly with minimal management effort or intervention from the service provider" [8].

Using a service-oriented cloud infrastructure has some advantages. First, the interoperability in independent programming language technologies, along with the means of transport provided by the HTTP protocol. It allows